

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Procesos Tecnológicos e Industriales

Sustentabilidad y Tecnología

PROYECTO DE APLICACIÓN PROFESIONAL (PAP)

PAP4G03 Programa de Apoyo a Centros de Investigación Externos



**ITESO, Universidad
Jesuita de Guadalajara**

**PAP4G03 PAP PROGRAMA DE APOYO A CENTROS DE INVESTIGACIÓN
EXTERNOS I**

**“Algoritmos cuánticos implementados en una computadora de diamante
C12-C13”, Centro Universitario de Ciencias Exactas e Ingenierías de la
Universidad de Guadalajara, Guadalajara, Jalisco**

PRESENTAN

Programas educativos y Estudiantes

Ing. en Nanotecnología – Karina Chiñas Fuentes

Ing. en Nanotecnología – Daniel Hernández Mota

Profesor PAP: Dr. Gustavo López Velázquez

Tlaquepaque, Jalisco, diciembre 2018

ÍNDICE

Contenido

REPORTE PAP	2
Presentación Institucional de los Proyectos de Aplicación Profesional	2
Resumen	2
1. Introducción	3
1.1. Objetivos	3
1.2. Justificación	4
1.3. Antecedentes	5
1.4. Contexto	6
2. Desarrollo	8
2.1. Sustento teórico y metodológico	9
2.2. Planeación y seguimiento del proyecto.....	22
3. Resultados del trabajo profesional	44
4. Reflexiones del alumno o alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto	51
5. Conclusiones	56
6. Bibliografía	58
Anexos	61

REPORTE PAP

Presentación Institucional de los Proyectos de Aplicación Profesional

Los Proyectos de Aplicación Profesional (PAP) son una modalidad educativa del ITESO en la que el estudiante aplica sus saberes y competencias socio-profesionales para el desarrollo de un proyecto que plantea soluciones a problemas de entornos reales. Su espíritu está dirigido para que el estudiante ejerza su profesión mediante una perspectiva ética y socialmente responsable.

A través de las actividades realizadas en el PAP, se acreditan el servicio social y la opción terminal. Así, en este reporte se documentan las actividades que tuvieron lugar durante el desarrollo del proyecto, sus incidencias en el entorno, y las reflexiones y aprendizajes profesionales que el estudiante desarrolló en el transcurso de su labor.

Resumen

Este escrito trata el proceso de investigación teórica enfocada en la simulación de una computadora cuántica basada en diamante dopado con isótopos de carbono-13, consideradas como partículas de *spin* 1/2. Para ello, se explican los principios básicos y más importantes de la teoría de la Mecánica Cuántica y se expone la lógica matemática que resuelven las ecuaciones que simulan los sistemas cuánticos; además, se presentan códigos de programación en PYTHON que resuelven numéricamente dichas ecuaciones.

El objetivo principal de esta investigación es simular los mismos sistemas cuánticos, pero con diferentes parámetros, y conocer el comportamiento de cada uno con el fin de encontrar una metodología cuasi-general para la generación de algoritmos cuánticos dependientes del número de partículas.

Durante esta investigación, se logró llegar a la simulación teórica de un sistema de hasta tres partículas que interaccionan de forma diferente bajo un campo magnético

variante en el tiempo. Para dicho sistema, se derivó a una metodología que simula el proceso de *entrelazamiento cuántico* y *teleportación cuántica*; lo que significa que se crearon las bases, de condiciones ideales, para una técnica de investigación tipo experimental para la formación de un algoritmo cuántico.

Finalmente, se describen los principales aprendizajes y logros que se obtuvieron de esta estancia de investigación, tanto académico como social. Tal como haber logrado un vínculo con ITESO y CUCEI para investigación en física teórica, ya que sólo se tenía para investigación experimental; así como la asistencia que logramos aportar a esta línea de investigación y al Doctor encargado.

1. Introducción

1.1. Objetivos

Este proyecto busca el desarrollo de destrezas dentro del área de computación cuántica para hacer frente a conflictos en la investigación y en la sociedad, fortaleciendo de esta manera el vínculo de colaboración existente entre los centros de investigación y los alumnos del ITESO. Aunado a esto, se persigue mejorar el desempeño de investigación agregando valor al nivel académico de ambas instituciones.

Las destrezas pertinentes por desarrollar en el proyecto son la elaboración de un programa en FORTRAN que pueda simular una computadora cuántica, basada en la estructura cristalina del diamante dopado con carbono-13, contribuyendo así al mejoramiento científico y tecnológico desde una perspectiva teórica.

Además de ello, se busca facilitar la accesibilidad a este tipo de información por lo que se realizará una redacción de notas en LATEX sobre computación e información cuántica, con el fin de que sean notas entendibles para las personas que buscan integrarse o introducirse al tema y así contribuir al desarrollo de la ciencia y

divulgación del conocimiento. Para ello, pensamos utilizar diferentes interfaces tales como GITHUB, JUPYTER y OVERLEAF, ya que en ellas es posible programar en los lenguajes mencionados y seguir trabajando en línea.

1.2. Justificación

El campo de la investigación científica desde el punto de vista teórico enfocado a la computación cuántica no tiene un impacto inmediato en la generación de tecnología o resolución de problemáticas sociales. En realidad, los avances y las contribuciones de este tema acercan cada vez la posibilidad de que se puedan solucionar, de una manera más eficiente, ciertos problemas de computación que son complejos sirviendo esto como un beneficio para la humanidad. Algunos de los conflictos que se buscan resolver en la aplicación de computadoras cuánticas son el descubrimiento de fármacos mediante la simulación de interacciones moleculares (1), reducción de tiempo en optimización de procesos y encriptación de información (2; 3). Sin embargo, la implementación de estos sistemas sigue estando limitada, por lo que el principal reto que se tiene que superar es determinar los sistemas físicos en los cuales se pueda trabajar la computación cuántica de manera eficaz. Hacer esto requiere de experimentación costosa y complicada de hacer, ya que las condiciones necesarias para una computadora cuántica son muy específicas, como mantener temperaturas cercanas al cero absoluto, manipular nanoestructuras con precisión y contar con alto vacío, es decir, una presión 10 billones de veces menor a la atmosférica (4). Es por ello que es importante llevar a cabo la investigación teórica, ya que puede prevenir un gasto de recursos (tales como de materiales, económicos o de tiempo) y, al mismo tiempo, se modela un comportamiento de los resultados como si se tratara de investigación práctica (5).

En consecuencia, la investigación enfocada al ámbito de la computación cuántica requiere de un acercamiento teórico para determinar la viabilidad del sistema y posteriormente un desarrollo experimental. Es por ello que es necesario tener

mecanismos de simulación dada la dificultad para obtener este tipo de sistemas físicos. Al simular, se crea una versión aproximada del modelo cuántico, para posteriormente decidir con qué valores, parámetros y condiciones óptimas se pueden trabajar. De esta forma, se genera un impacto beneficioso y directo en los centros de investigación, ya que la investigación teórica conlleva una reducción de recursos para el estudio del tema en cuestión.

De manera local, se pueden obtener beneficios que no están ligados a la computación cuántica pero sí a la investigación y a la cultura científica. Al contribuir a este tipo de investigaciones para generar información, se abren las puertas a las personas que buscan introducirse o involucrarse en estos tópicos, desarrollando en paralelo un acercamiento mayor al desarrollo científico-tecnológico que funge como herramienta para ayudar a la sociedad. La divulgación y el acceso de información genera una transformación en la cultura con el objetivo de cambiar paulatinamente las reglas de conducta, los medios y las tecnologías usadas para dar una solución de diversos problemas (6).

Por otra parte, ITESO tiene un beneficio directo debido a que, con el conocimiento adquirido, se puede introducir el tema de computación cuántica a cursos como Física Cuántica II, Simulación Matemática e inclusive Álgebra Lineal. Esto es factible debido a que este tema genera un mejor entendimiento sobre el alcance de la física cuántica y las matemáticas. Esto mejora la calidad de educación ya que se puede comenzar a abordar un tema de especialización, relativamente complejo, desde la licenciatura. Complementario a esto, junto con la ayuda de la Sociedad de Alumnos de Nanotecnología de ITESO, es posible realizar pláticas de cómo fue nuestra experiencia PAP en investigación teórica para que los estudiantes de la carrera tengan una visión amplia y un mejor panorama de lo que podrían elegir como opción en investigación o trabajo.

1.3 Antecedentes

El grupo de investigación encuadrando el área respectiva de la Información y Computación Cuántica está conformado por dos integrantes, el Dr. Andrei Klimov y el Dr. Gustavo López Velázquez; ambos siendo investigadores Nivel II del Sistema Nacional de Investigadores y profesores investigadores titulares de la Universidad de Guadalajara. El Dr. Klimov se encarga de la investigación en Óptica e Información Cuántica mientras que el Dr. López se enfoca en el tópico de Computación Cuántica perteneciendo todas estas ramas a la Física fundamental avanzada.

En el año 2002 se instauró el proyecto de Computación Cuántica en un Sistema en Cadena con Átomos Paramagnéticos de Espín (spin) Nuclear de $\frac{1}{2}$ en colaboración con la Institución Los Alamos National Laboratory (LANL). No obstante, este proyecto recibió muy poco financiamiento debido a la problemática general que se tiene en las computadoras cuánticas: el sistema no es escalable y también el ruido externo produce errores en el funcionamiento, así como la pérdida de propiedades cuánticas de los estados entrelazados. Después de hacer las simulaciones, durante 10 años, se pudo determinar que este modelo no era un sistema cuántico viable para realizar físicamente. Por lo que, en el año 2012, el Dr. López consideró independientemente un modelo alternativo, el cual está compuesto por átomos de carbono en forma alotrópica de diamante dopados con el isótopo C13 debido a que este también cuenta con la propiedad de tener spin de $\frac{1}{2}$.

Antes de formar una colaboración con el ITESO para poder llevar a cabo el Proyecto de Aplicación Profesional en esta área, se estaban definiendo los parámetros del modelo a utilizar con el *algoritmo cuántico de Shor*. Hoy en día es necesario implementar algoritmos que en principio simulen el sistema cuántico y con esto comenzar a aplicar operadores para determinar su evolución en el tiempo.

1.4. Contexto

Desgraciadamente, factores de la historia de México, tales como el Porfiriato, la Revolución y sus periodos de limitar la educación a clases sociales privilegiadas, causaron que hubiera un retraso en el impulso científico en el país (7). De hecho, durante casi un siglo (desde el año 1820 hasta 1929) no ocurrió ninguna relación importante entre la Ciencia y el Estado debido a inestabilidades y fenómenos sociales las cuales impidieron que la ciencia y la tecnología coadyuvaran de forma importante y sistemática al desarrollo nacional. Después del año 1929, comenzó a haber un ambiente para promover la investigación científica, no obstante, aún después de once décadas, todavía no se cuenta con un sistema de investigación y desarrollo tecnológico útil para fortalecer el PIB (Producto Interno Bruto) (8). Estos rezagos en la historia hacen que hoy en día existan varios retos para tener un buen desarrollo científico y tecnológico. Entre estos retos se encuentra llevar a cabo una mejora a diversos factores como la divulgación científica, la inversión, la infraestructura, el vínculo entre la industria y la academia, las políticas nacionales y los recursos humanos.

A pesar de que el porcentaje del PIB, que se invierte en gastos enfocados a la investigación y el desarrollo, ha ido aumentando de 0.26% en 1996 a 0.53% en el 2014, no fue suficiente para tratar los diversos conflictos que se vivían y viven en México. En el 2015 se registró un valor de 0.552 del %PIB destinado a esta área, porcentaje menor que el promedio de América Latina y el Caribe, con cifras de 0.767%, posicionándonos en el lugar número 32 de 74 países que registraron sus datos para el mismo año (9). En los datos registrados del mismo año se encuentran valores que van de 0.04% a 4.27%.

Por otra parte, en el año 2017 el CONACYT (Consejo Nacional de Ciencia y Tecnología) sufrió un recorte del 23% en referencia a su presupuesto otorgado el 2016, causando que el dinero fuera insuficiente para sostener los diversos programas de becas con los que cuenta; incluso afirmaron que no habría beca alguna para diversos encabezados de posgrados para ese año en curso (10). Para el año 2018, se le volvió a reducir con un 3.7% respecto a lo asignado para el año

2017; se cree que uno de los principales motivos ha sido el deterioro del valor de la moneda gracias a la inflación (11). Este recorte afectó a todos los campos de investigación científica en México, perturbando por lo tanto al grupo de Investigación enfocada a la Computación e Información Cuántica.

En México, los grupos que hacen investigación en este campo son muy reducidos. Existe la División de Información Cuántica (DICU) de la Sociedad Mexicana de Física, donde el Dr. Klimov es miembro, la cual ha mostrado actividades de integración como oferta de becas patrocinadas por diversas instituciones en Óptica Cuántica, Materia Fría y e Información Cuántica (12). Referente a la Computación Cuántica, hay un grupo que se dedica específicamente al desarrollo de Algoritmos Cuánticos, el cual está siendo dirigido por el Dr. Salvador E. Venegas-Andraca, quien también es miembro de la DICU y ha escrito varios documentos respecto a Computación Cuántica, Algoritmos Cuánticos, Caminatas Cuánticas y Mecánica Cuántica. Pero en contraste con las instituciones que se encargan de investigar, todavía no hay empresas que estén implementando esta tecnología.

Respecto a la Computación Cuántica en otros países, existen diversas empresas privadas y empresas emergentes que ven por su desarrollo y la aplicación para la solución de problemas. Algunos de estas empresas son IBM, Rigetti Computing, 1Qbit, D-Wave y QxBranch, quienes tratan temas de *Machine Learning*, *hardware* y *software* cuántico, algoritmos cuánticos, simulaciones de sistemas cuánticos, análisis de riesgo y problemas financieros. Por ejemplo, una de las empresas emergentes, QxBranch, logró recabar \$8,468,722 dólares estadounidenses en un periodo de 4 años (13). Además de ello, algunos de los gobiernos que han mostrado interés en su desarrollo han sido Reino Unido, la Unión Europea y Estados Unidos (14).

2. Desarrollo

La computación cuántica es un tema que requiere de fundamentos teóricos basados en la mecánica cuántica y computación. Por ello, en esta sección, se tratarán conceptos elementales que hacen posible esta investigación, así como la metodología necesaria para su realización y los avances que han obtenido durante el semestre.

2.1. Sustento teórico y metodológico

Para comenzar a entender la computación cuántica, es necesario remontarnos a las bases de la mecánica cuántica. Como tal, la mecánica cuántica es una teoría matemática que describe el comportamiento e interacción de fenómenos atómicos y subatómicos también conocidos como sistemas cuánticos (SC). Los SC tienen escalas dimensionales de Angstroms¹, generando efectos que no se observan en sistemas clásicos, por lo que utilizar las ecuaciones de movimiento clásicas no es suficiente para poder describir completamente el comportamiento de los fenómenos clásicos (15).

Cuando se trabaja con SC, el principal objetivo es obtener la *función de onda o estado* Ψ de dicho sistema. La función de onda se obtiene al resolver la *ecuación de Schrödinger* (mostrada en la ecuación 2.1)

$$i\hbar \frac{\partial \Psi}{\partial t} = \frac{\hat{P}^2}{2m} \Psi + V(\mathbf{r}, t) \Psi \quad [2.1]$$

Donde \mathbf{r} es la posición en el espacio tridimensional, t es el tiempo, m hace referencia a la masa de la partícula en el análisis, i es el número complejo $\sqrt{-1}$ (notar que esto indica que Ψ es una función compleja²), \hat{P} es el operador momento y $V(\mathbf{r}, t)$ es la energía potencial al que se encuentra sometido el sistema (16). Otra manera de encontrar a la ecuación 2.1 es de la forma

¹ Unidad de medición referente a longitud; denotado por el símbolo Å, donde $1\text{Å} = 10^{-10}$ m.

² Un complejo se expresa como $z = a + ib$; a es la parte real y b es la parte imaginaria de z .

$$i\hbar \frac{\partial \Psi}{\partial t} = \hat{H} \Psi \quad [2.2]$$

donde \hat{H} es el operador *Hamiltoniano*, describe la evolución del sistema dada la energía en el mismo, esto es porque el operador Hamiltoniano se define como

$$\hat{H} = \frac{\hat{P}^2}{2m} + V(\mathbf{r}, t) \quad [2.3]$$

Es decir que se define como la suma de la energía cinética y potencial, pero en forma de operadores. Por otra parte, es vital saber que la magnitud al cuadrado de la función de onda representa la densidad de probabilidad del sistema, eso implica

$$\int_{\varepsilon} |\Psi(\mathbf{r}, t)|^2 d^3x = 1 \quad [2.4]$$

que significa que la probabilidad de encontrar a la partícula en todo el espacio ε debe ser igual a 1. Esto es importante porque la ecuación 2.4 nos dice que no toda solución de la ecuación 2.1 nos da la descripción del SC en cuestión, por lo que todo sistema físicamente realizable debe de contar con una función de onda tipo cuadrado integrable (16).

Para esta investigación, se utilizará la *notación de Dirac*, también conocida como notación Bra-Ket; en esencia, la notación de Dirac aborda un análisis de la función de onda como un vector columna que pertenece al *espacio de Hilbert*³; por lo que, la ecuación 2.2 tomaría la forma de

$$i\hbar \frac{\partial |\Psi\rangle}{\partial t} = \hat{H} |\Psi\rangle \quad [2.5]$$

donde $|\Psi\rangle$ se lee como el vector ket del estado Ψ . Mientras que la ecuación 2.4 se ve reducida como

³ Espacio complejo completo de dimensión finita o infinita cuyo contenido se caracteriza por ser cuadrado integrable (25).

$$\langle \Psi | \Psi \rangle = 1 \quad [2.6]$$

En esta expresión, $\langle \Psi |$ representa al vector transpuesto conjugado de $|\Psi\rangle$, el cual se lee como el vector bra del estado Ψ (17). La ecuación 2.6 muestra el producto interno del estado Ψ con el Ψ ; este producto interno representa la *proyección* del estado del vector en bra sobre el estado del vector en ket. Aparte de esto, es importante mencionar que el Hamiltoniano es un operador *observable*, esto quiere decir que es *hermítico*⁴, por lo que cuenta con ciertas propiedades importantes: sus eigenvalores son de carácter real y sus eigenvectores son ortogonales (17).

Con estos fundamentos, desde una perspectiva matemática, se puede generar un SC específico el cuál se define como un set de dos estados arbitrarios con una base $\{|0\rangle, |1\rangle\}$ para que el vector $|\Psi\rangle$ pueda ser representado como:

$$|\Psi\rangle = C_0 |0\rangle + C_1 |1\rangle \quad [2.6]$$

A este SC se le conoce como bit cuántico (*qubit* por sus siglas en inglés). Esto se da debido a que es un objeto matemático que tiene solamente dos estados, por lo que se puede utilizar para guardar de alguna manera información y para computar. La información almacenada se puede ver reflejada en los coeficientes complejos correspondientes al estado que se está analizando y se puede modificar al aplicar algún potencial variante (18). Por lo tanto, para determinar los coeficientes respectivos se tiene que hacer uso de la ecuación de Schrödinger. Este principio básico es lo que hace que la computación cuántica sea posible. El análogo clásico se puede pensar como el bit haciendo referencia a un estado cero o un estado uno; pero una de las diferencias más notables entre el sistema clásico y el cuántico, es que el qubit puede tener una superposición de los estados debido a su naturaleza matemática. No obstante, para poder tener información acerca del qubit es

⁴ El operador y su transpuesto conjugado son equivalentes.

necesario hacer una *medición*, lo cual causaría el *colapso de la función*⁵ de onda a uno de los dos estados. Por lo que lo único que se observaría sería un bit clásico con cierta probabilidad de ser cero o uno, perdiendo así información adicional del SC (18), o manipulando de manera estratégica para crear algoritmos prácticos.

Para representar un qubit se utiliza el concepto llamado la esfera de Bloch (véase la **Figura 1**). Esto se puede lograr modificando los coeficientes C_0 y C_1 de la ecuación 2.6 por una selección dónde se utilizan ciertos ángulos para describir el sistema propiamente desde el punto de vista esférico (19). La ecuación que representa la esfera de Bloch es

$$|\Psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle \quad [2.7]$$

La ecuación 2.7 permite que el sistema cuántico se pueda implementar como una esfera, ya que se cuentan con distintos ángulos con los cuales se puede ubicar algún punto en su superficie. Es fácil determinar que la magnitud al cuadrado de este SC siempre es 1. Por ende, la esfera de Bloch se utiliza para apreciar geoméricamente los estados del SC, es decir de un qubit. Sin embargo, cabe mencionar que esta descripción está limitada a el uso de un solo qubit.

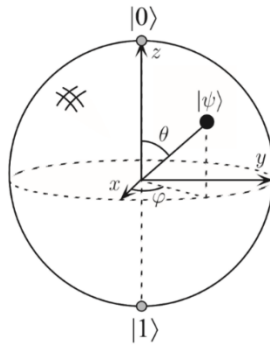


Figura 1. Esfera de Bloch. Visualmente, el estado $|\Psi\rangle$ se puede determinar gracias al valor de los ángulos θ y φ , con esto se puede determinar cualquier estado de un qubit (19).

⁵ La función de onda deja de ser cuadrado integrable (17).

Para el caso de múltiples qubits, se utiliza la función de onda para la descripción del SC. En específico para dos qubits, se tiene una base de cuatro estados debido a las posibles combinaciones que se pueden obtener de los estados individuales, por lo que la función de onda queda de la siguiente manera:

$$|\Psi\rangle = C_{00} |00\rangle + C_{01} |01\rangle + C_{10} |10\rangle + C_{11} |11\rangle \quad [2.8]$$

Para este y los siguientes casos se siguen cumpliendo todas las condiciones de normalidad de los sistemas cuánticos. Se puede obtener una expresión generalizada para un sistema de n qubits empleando una notación de sumatoria. En una base compuesta por los n qubits expresada como $\{|\alpha\rangle = |\alpha_n \dots \alpha_1\rangle\}$ para cada valor definido como $\alpha_i = 0,1$ se tiene un espacio de Hilbert con dimensión igual a 2^n . El SC correspondiente queda de la forma

$$|\Psi\rangle = \sum_k^{2^n} C_k |\alpha\rangle \quad [2.9]$$

Al hacer la expansión de sumatoria para cualquier n , se puede obtener la ecuación que representa cualquier estado.

Para poder manipular un qubit, es necesario utilizar compuertas lógicas; es mediante el uso de operadores unitarios los que hacen posible este manejo. Se pueden usar distintas compuertas para crear los algoritmos de computación cuántica. Una de las características importantes de las mismas es que son reversibles y, por lo tanto, no hay pérdida de energía al usarlas (19). Como se muestra en la **Figura 2**, cada compuerta tiene una matriz asociada, lo cual ayuda a determinar el efecto que tiene utilizarlas en el qubit. En concreto, se puede determinar el efecto de las compuertas actuando sobre un estado puro arbitrario $|\alpha\rangle$.

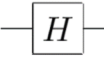
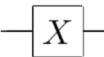
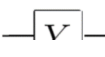
Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli-X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$

Figura 2. Compuertas de un qubit. En los circuitos cuánticos son representadas por los símbolos encontrados en el centro de la imagen, a la derecha se puede apreciar la matriz asociada a la compuerta cuántica. Se evalúa la acción de las compuertas actuando sobre los dos estados del qubit y observando cómo es modificado (19).

Para la compuerta Hadamard se tiene que se genera una superposición de los dos estados, si $\alpha = 0$, entonces se le aplica el operador Hadamard al estado base y se obtiene la superposición con la suma de ambos estados. En caso de que $\alpha = 1$, se obtiene también una superposición, pero con una resta del coeficiente del estado $|1\rangle$.

$$\hat{H} |\alpha\rangle = \frac{1}{\sqrt{2}} (|0\rangle + (-1)^\alpha |1\rangle) \quad [2.10]$$

La compuerta Pauli-X, también conocida como X o NOT se tiene la negación del estado que se tenía, es decir si se tiene 0 cambia a 1 o viceversa, por lo que se utiliza el símbolo de suma modulo dos⁶ \oplus .

$$X |\alpha\rangle = |\alpha \oplus 1\rangle \quad [2.11]$$

⁶ Operaciones que regresan al valor inicial después de alcanzar cierto valor crítico llamado módulo. Para el caso del módulo 2, se tienen solamente los valores 0 y 1, donde $0 + 1 = 1$, pero $1 + 1 = 0$. Se puede entender por analogía a las horas del reloj.

Para la compuerta Pauli-Y o Y, se tiene la negación multiplicada por un coeficiente imaginario con otro factor que lo pudiera hacer negativo (si $\alpha = 0$ queda positivo, en caso contrario con $\alpha = 1$ queda negativo).

$$Y |\alpha\rangle = (-1)^\alpha i |\alpha \oplus 1\rangle \quad [2.12]$$

Para la compuerta Pauli-Z o Z, si $\alpha = 0$, no ocurre un cambio, no obstante, en el caso de que $\alpha = 1$ se obtiene el negativo del estado $|1\rangle$.

$$Z |\alpha\rangle = (-1)^\alpha |\alpha\rangle \quad [2.13]$$

La compuerta fase no tiene efecto si se tiene que $\alpha = 0$, pero si $\alpha = 1$, se aplica la acción de multiplicar por la unidad imaginaria el estado $|1\rangle$.

$$S |\alpha\rangle = e^{i\frac{\pi}{2}\alpha} |\alpha\rangle \quad [2.14]$$

La compuerta $\frac{\pi}{8}$ es la raíz de la compuerta fase, por lo que los resultados son parecidos, pero en el caso de $\alpha = 1$, se obtiene la multiplicación al estado $|1\rangle$ por la raíz de la unidad imaginaria.

$$T |\alpha\rangle = e^{i\frac{\pi}{4}\alpha} |\alpha\rangle \quad [2.15]$$

No obstante, estas compuertas se aplican solamente a un solo qubit. Por lo que, si se quieren hacer cambios en sistemas cuánticos con más qubits, se deben de aplicar compuertas que tienen una representación matricial de mayor dimensión. Para dos qubits, la compuerta más utilizada para desarrollar algoritmos cuánticos es la compuerta llamada *Controlled-NOT* también representada por CNOT o CN (ver la **Figura 3**). Esta compuerta relaciona dos qubits y hace una acción dependiendo del primero; en específico si el primer qubit está en el estado $|0\rangle$, no aplica nada al segundo, en cambio, si el primer qubit está en el estado $|1\rangle$, aplica

una compuerta NOT (19), es por esta razón el nombre de “controlled”: depende del primer qubit para hacer la acción.

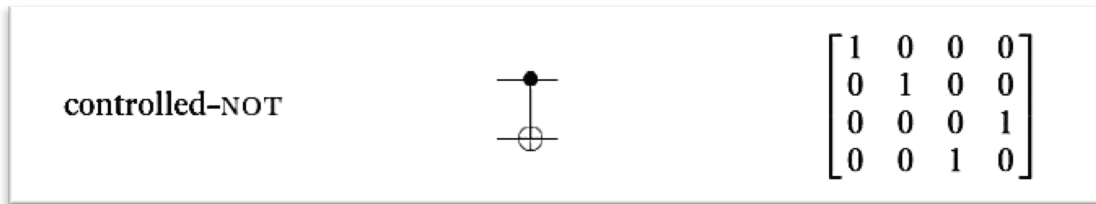


Figura 3. Compuerta CNOT. Esta compuerta recibe dos qubits, al primero no lo modifica mientras que el segundo es afectado dependiendo del valor del primer qubit. Esto genera así mismo una correlación entre los dos qubits (19).

Esta compuerta puede actuar en un SC de n qubits, afectando solamente dos qubits específicos, por esto se recomienda utilizar una notación con índices, dónde la acción sobre los elementos “ i ” y “ j ” queda determinada por:

$$CN_{ij} |\alpha\rangle = CN_{ij} |\alpha_n \dots \alpha_i \dots \alpha_j \dots \alpha_1\rangle = |\alpha_n \dots \alpha_i \dots (\alpha_j \oplus \alpha_i) \alpha_{j-1} \dots \alpha_1\rangle \quad [2.16]$$

Cabe destacar que cualquier compuerta lógica de qubits múltiples puede ser compuesta por la compuerta CNOT y las compuertas lógicas simples es decir que cualquier algoritmo cuántico se puede descomponer hasta tener una base de estas compuertas básicas, por lo que se consideran que estos operadores son la base canónica para generar cualquier otro operador (20).

Al aplicar la compuerta Hadamard y la compuerta CNOT en orden específico, es posible crear sistemas cuánticos con propiedades interesantes, conocidos como los estados de Bell o pares EPR, (llamados así porque Einstein, Podolsky y Rosen describieron su comportamiento) (véase la **Figura 4**).

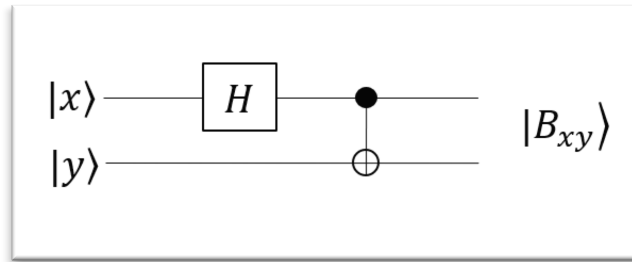


Figura 4. Circuito que genera los estados de Bell (21). Los valores x y y son estados puros con posibles valores $|0\rangle$ o $|1\rangle$. Posteriormente al estado x se le aplica una compuerta Hadamard y se aplica un CNOT a y . Esto genera los estados de Bell $|B_{xy}\rangle$ que están intrínsecamente entrelazados (19).

Los estados de Bell cumplen con la propiedad en la cual es posible determinar el estado de un qubit función sin necesidad de colapsarlo, solamente viendo uno de los dos qubits. En total, existen cuatro estados de Bell, y se pueden generar dependiendo de los valores que se introduzcan, estos se pueden observar en la **Figura 5**.

$$\begin{aligned}
 |\beta_{00}\rangle &= \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \\
 |\beta_{01}\rangle &= \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle) \\
 |\beta_{10}\rangle &= \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle) \\
 |\beta_{11}\rangle &= \frac{1}{\sqrt{2}} (|01\rangle - |10\rangle)
 \end{aligned}$$

Figura 5. Estados de Bell (19).

Para poder observar las propiedades de los estados de Bell tomaremos como ejemplo el primer estado de la **Figura 5**, $|\beta_{00}\rangle$. Al tener este estado se observa el primer qubit y se tiene 50% de probabilidad (obtenida de la magnitud a cuadrado del coeficiente correspondiente) de que el resultado sea que se encuentre en el

estado $|1\rangle$ o $|0\rangle$. Al definir este estado, automáticamente el otro qubit se define, sin la necesidad de también observarlo. Por ejemplo, en caso de que salga el estado $|1\rangle$, se sabe que el estado de ambos qubits se encuentra definido como $|11\rangle$ indicando que el segundo qubit también está en el estado $|1\rangle$. En caso de que se hubiera obtenido el estado $|0\rangle$ del primer qubit, se sabe automáticamente que se está en el estado $|00\rangle$ y por lo tanto el estado del segundo qubit queda determinado como $|0\rangle$. Este mismo razonamiento se puede generalizar de los demás estados de Bell.

Debido a que es posible determinar un estado a partir del otro, se dice que estos dos estados cuánticos están correlacionados, a este fenómeno se le conoce como entrelazamiento cuántico (20). En realidad, la medición de un estado entrelazado implica el resultado de la otra debido a esta correlación que se tiene; esta propiedad cuántica brinda información acerca del estado del sistema de manera instantánea ya que, al estar entrelazados, no se puede determinar el estado de un qubit de manera separada, es decir, que el estado de cada qubit está sujeto a su propio estado individual como al estado en conjunto con el otro qubit (18). Esto se debe a la interacción entre los qubits en el SC así como a la compuerta que los relaciona (20) (la interacción entre los qubits y la compuerta se toma en cuenta como una energía de interacción que también modifica la ecuación de Schrödinger).

Para poder llevar a cabo la realización del sistema físico, se pueden utilizar varias propuestas para generar qubits; entre estas están la polarización de la luz, uso de orbitales atómicos, spin nuclear, o valores energéticos de una partícula sometida a un potencial dado (18). El único requisito es que es necesario mantener dos niveles energéticos por partícula y así mantener la propiedad binaria. Para esta investigación, se utilizó el sistema que hace uso del spin nuclear de las partículas. En este caso se utilizan fermiones⁷ con valor de spin de un medio, lo cual permite generar un sistema que solamente se mantiene con dos estados (spin arriba y spin

⁷ Partículas de spin semi-entero.
FO-DGA-CPAP-0017

abajo) y por lo tanto es suficiente para representar un qubit. Debido a esto, el sistema físico utilizado consta de una estructura cristalina del diamante (véase **Figura 6**) compuesta en general de carbono 12 (C12), dónde se cambian N átomos alineados en una misma dirección (eje Z) por isótopos de carbono 13 (C13). Esto se puede observar en la **Figura 7**. Los átomos de C13 tienen un spin $\frac{1}{2}$, en comparación con los átomos de C12, con spin 0.

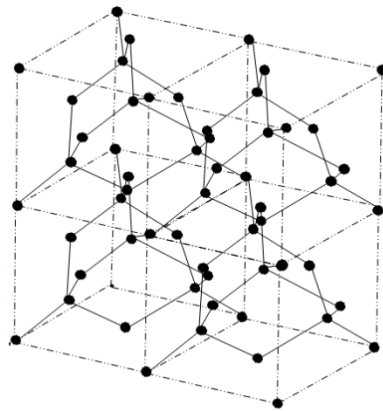


Figura 6. Estructura cristalina del diamante. Consta de una estructura por ser cúbica centrada en las caras (CCC) (21).

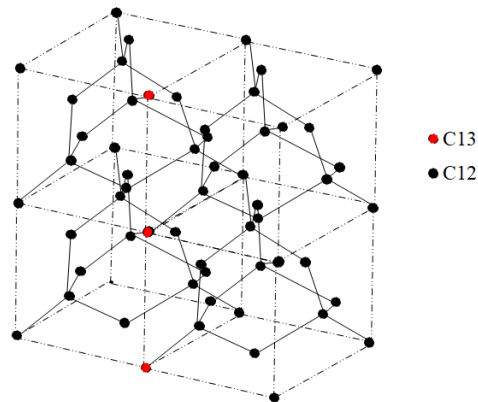


Figura 7. Sistema físico utilizado. En esta estructura de diamante, se sustituyeron tres átomos de C12 por otros de C13. Debido al spin que tienen estas partículas, se pueden aplicar campos magnéticos para interactuar con ellos (21).

Para entender la razón de utilizar este sistema, es importante comprender el spin. En términos coloquiales, el spin se puede interpretar como una descripción del

momento angular intrínseco que funge como una rotación en el propio eje⁸ de la partícula (18). Determinando el valor del momento angular total, y restringiendo a las partículas de tener momento lineal (y por ende eliminar el momento angular orbital), se puede hacer una equivalencia de que el momento angular total estaría dado simplemente por el spin. El momento angular total está ligado al momento magnético descrito por

$$\vec{m} = \gamma \vec{J} = \gamma (\vec{L} + \vec{S}) = \gamma \vec{S} \quad [2.16]$$

Donde \vec{m} es el momento magnético, γ es la constante giromagnética, \vec{J} es el momento angular total, \vec{L} es el momento angular orbital, \vec{S} es el spin (o momento angular intrínseco). Es importante aclarar que el spin se puede considerar como un vector operador, por lo que tiene componentes en los ejes “x”, “y”, y “z”⁹. Así mismo se pueden tener otros operadores relacionados que se reconocen como operadores escaleras del spin. Estos operadores actúan de manera importante debido a que tienen una acción directa ante los estados del SC.

Los operadores escalera se definen como $S_+ = S_x + iS_y$ y $S_- = S_x - iS_y$. La acción sobre los dos estados del sistema cuántico se puede observar en las siguientes ecuaciones.

$$S_+ |0\rangle = 0, \quad S_+ |1\rangle = \hbar |0\rangle \quad [2.17]$$

$$S_- |0\rangle = \hbar |1\rangle, \quad S_- |1\rangle = 0 \quad [2.18]$$

Determinar este comportamiento es crítico debido a que en la investigación se hace uso de estos operadores cuando se desarrolla la ecuación de Schrödinger. Al tener

⁸ Estrictamente, la idea de rotación viene de una perspectiva clásica, es errónea desde el punto de vista cuántico debido a que no se carece de sentido hablar de la rotación de una partícula desde que esta no se visualiza como un cuerpo sino como un evento físico que sigue el comportamiento de una función de onda probabilística. No obstante, para los propósitos del escrito, se menciona para que el concepto se pueda entender con facilidad.

⁹ Cada componente tiene una matriz asociada, las matrices están dadas por las matrices de Pauli.

un momento magnético, se tiene un término de energía asociado, U , a un campo magnético externo \vec{B} dada por la expresión

$$U = -\vec{m} \cdot \vec{B} = -\gamma \vec{S} \cdot \vec{B} \quad [2.19]$$

Para tomar en cuenta el spin en la ecuación de Schrödinger es necesario hacer algunas modificaciones ya que se incorporan estas componentes del spin en forma de las matrices de Pauli σ , obteniendo la ecuación de Pauli¹⁰

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = \left(\frac{(\vec{\sigma} \cdot \hat{p})^2}{2m} + V(x, t)\mathbb{I} \right) |\psi\rangle \quad [2.20]$$

Es importante notar que al forzar que la partícula esté localizada (es decir, que se encuentre fija gracias al enlace atómico en la red del diamante), deja de tener momento lineal y la ecuación de Schrödinger se simplifica, facilitando su resolución; Por ende, al no haber una cantidad de movimiento e incorporar la energía, se tiene solamente que contar el potencial que está compuesto por la energía asociada al campo magnético externo de la ecuación 2.19, y así mismo, la interacción que existe entre los qubits (U_{int}) del SC. Por lo que la expresión de la ecuación de Pauli queda reducida a

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = V(x, t) |\psi\rangle = \left(-\gamma \vec{S} \cdot \vec{B} + U_{int} \right) |\psi\rangle \quad [2.21]$$

Esta ecuación fue la que se utilizó para describir el comportamiento de los estados del qubit debido a que solamente se modifican los coeficientes complejos ya que surgen cambios a través del tiempo y por ende se ven afectadas las probabilidades.

¹⁰ Notablemente similar a la ecuación de Schrödinger, sólo es válida para fermiones con valor de spin = 1/2.

2.2. Planeación y seguimiento del proyecto

- *Descripción del proyecto*

La principal tarea realizada para esta línea de investigación fue encontrar las soluciones de la ecuación de Schrödinger (como la ecuación 2.21), y observar la dependencia temporal de los coeficientes de las funciones de onda. Esto se hace para determinar las condiciones necesarias en las cuales el potencial implementado genera una compuerta cuántica dependiendo del tiempo que se utilizó, así como los parámetros de frecuencia del campo magnético, y por ende permitiendo poco a poco el desarrollo de algoritmos que hacen posible la computación cuántica. No obstante, hacer esto de forma analítica puede llegar a ser muy difícil y tardado ya que se llega a un sistema de ecuaciones diferenciales acopladas. Por esta razón, se opta por encontrar las soluciones mediante métodos numéricos. El método numérico que se decidió trabajar es Runge-Kutta de cuarto orden.

Utilizar cualquier método numérico implica definir un intervalo $\Delta t = t_f - t_i$ para el cual se va a resolver y definir un tamaño de paso h que recorrerá dicho intervalo, la relación entre ellos es a través del número de iteraciones m del algoritmo

$$m = \frac{t_f - t_i}{h} \quad [2.22]$$

En nuestro caso Δt es el intervalo de tiempos (en segundos) que buscamos graficar, mientras que el parámetro de h por lo general va de 0.1 a 0.001¹¹. Debido a que los coeficientes de las funciones de onda son complejos, estas van a tener su parte real y su parte imaginaria, por lo que para un sistema que da N ecuaciones¹², habrá $2N$ ecuaciones relacionadas entre sí a solucionar con Runge-Kutta. La convención con la que estaremos trabajando para desarrollar el método es que la parte real del

¹¹ Una h pequeña da mejores resultados numéricos, pero hace que el programa tarde más tiempo en correr; debido a esto, el valor de h se define a conveniencia.

¹² N estará definido por el número de qubits que estaremos trabajando, eso es: $N = (2)^{\#qubits}$.

coeficiente i -ésima¹³ estará dada por $X^{(i)}(t)$ y la parte imaginaria del coeficiente i -ésima estará dada por $Y^{(i)}(t)$. Por lo que, las ecuaciones $\dot{X}^{(i)}$ y $\dot{Y}^{(i)}$ que se tendrán a resolver tomarán la forma de

$$\frac{dX^{(i)}}{dt} = \dot{X}^{(i)} \left(t, X^{(j)}, Y^{(j)}, \dots, X^{(k)}, Y^{(k)} \right) \quad [2.23]$$

$$\frac{dY^{(i)}}{dt} = \dot{Y}^{(i)} \left(t, X^{(j)}, Y^{(j)}, \dots, X^{(k)}, Y^{(k)} \right) \quad [2.24]$$

donde, $i \neq j \neq k$. Runge-Kutta de cuarto orden consiste en utilizar las condiciones iniciales del sistema, en nuestro caso $X^{(i)}(0)$ y $Y^{(i)}(0)$, para calcular $8N$ constantes, que ayudarán a encontrar una aproximación a las soluciones de las N ecuaciones; ya que las aproximaciones de $X^{(i)}(t)$ y $Y^{(i)}(t)$ están dadas por (22)

$$X^{(i)}(t+h) = X^{(i)}(t) + \frac{1}{6} \left(K_1^{(i)} + 2K_2^{(i)} + 2K_3^{(i)} + K_4^{(i)} \right) \quad [2.25]$$

$$Y^{(i)}(t+h) = Y^{(i)}(t) + \frac{1}{6} \left(L_1^{(i)} + 2L_2^{(i)} + 2L_3^{(i)} + L_4^{(i)} \right) \quad [2.26]$$

Las constantes $K_1^{(i)}, K_2^{(i)}, K_3^{(i)}, K_4^{(i)}$ y $L_1^{(i)}, L_2^{(i)}, L_3^{(i)}, L_4^{(i)}$ se calculan con (22)

$$K_1^{(i)} = \dot{X}^{(i)} \left(t, X^{(j)}, Y^{(j)}, \dots, X^{(k)}, Y^{(k)} \right) \quad [2.27]$$

$$L_1^{(i)} = \dot{Y}^{(i)} \left(t, X^{(j)}, Y^{(j)}, \dots, X^{(k)}, Y^{(k)} \right) \quad [2.28]$$

$$K_2^{(i)} = h\dot{X}^{(i)} \left(t + \frac{h}{2}, X^{(j)} + \frac{K_1^{(j)}}{2}, Y^{(j)} + \frac{L_1^{(j)}}{2}, \dots, X^{(k)} + \frac{K_1^{(k)}}{2}, Y^{(k)} + \frac{L_1^{(k)}}{2} \right) \quad [2.29]$$

$$L_2^{(i)} = h\dot{Y}^{(i)} \left(t + \frac{h}{2}, X^{(j)} + \frac{K_1^{(j)}}{2}, Y^{(j)} + \frac{L_1^{(j)}}{2}, \dots, X^{(k)} + \frac{K_1^{(k)}}{2}, Y^{(k)} + \frac{L_1^{(k)}}{2} \right) \quad [2.30]$$

$$K_3^{(i)} = h\dot{X}^{(i)} \left(t + \frac{h}{2}, X^{(j)} + \frac{K_2^{(j)}}{2}, Y^{(j)} + \frac{L_2^{(j)}}{2}, \dots, X^{(k)} + \frac{K_2^{(k)}}{2}, Y^{(k)} + \frac{L_2^{(k)}}{2} \right) \quad [2.31]$$

¹³ n es un entero positivo que va de 1 a N .

$$L_3^{(i)} = h\dot{Y}^{(i)} \left(t + \frac{h}{2}, X^{(j)} + \frac{K_2^{(j)}}{2}, Y^{(j)} + \frac{L_2^{(j)}}{2}, \dots, X^{(k)} + \frac{K_2^{(k)}}{2}, Y^{(k)} + \frac{L_2^{(k)}}{2} \right) \quad [2.32]$$

$$K_4^{(i)} = h\dot{X}^{(i)} \left(t + h, X^{(j)} + K_3^{(j)}, Y^{(j)} + L_3^{(j)}, \dots, X^{(k)} + K_3^{(k)}, Y^{(k)} + L_3^{(k)} \right) \quad [2.33]$$

$$L_4^{(i)} = h\dot{Y}^{(i)} \left(t + h, X^{(j)} + K_3^{(j)}, Y^{(j)} + L_3^{(j)}, \dots, X^{(k)} + K_3^{(k)}, Y^{(k)} + L_3^{(k)} \right) \quad [2.34]$$

Es importante notar que la forma en la que están presentadas sigue el orden en el que se deben calcular debido a su mutua dependencia entre ellas; además, todas dependen del parámetro t , haciendo que las $8N$ constantes estén cambiando por cada iteración.

La **Figura 8** muestra el proceso lógico que seguiremos para solucionar las ecuaciones que lleguen a emerger de los diferentes sistemas cuánticos con los que estaremos trabajando.

Es importante mencionar que Runge-Kutta de cuarto orden no es el único método numérico con el que se puede resolver sistema de ecuaciones diferenciales, existen otros tales como el método del Euler, el método del punto medio, Runge-Kutta de segundo orden, entre otros; la razón por la cual se decidió trabajar con Runge-Kutta de cuarto orden fue porque es el que menor porcentaje de error carga en cada iteración. Por otra parte, el método de Euler consiste en calcular las derivadas de las ecuaciones que se buscan resolver, mientras más derivadas de mayor orden se calculen, el proceso se vuelve más preciso; en cambio, Runge-Kutta de cuarto orden se vuelve preciso al hacer más pequeño el tamaño de paso h para calcular constantes (23).

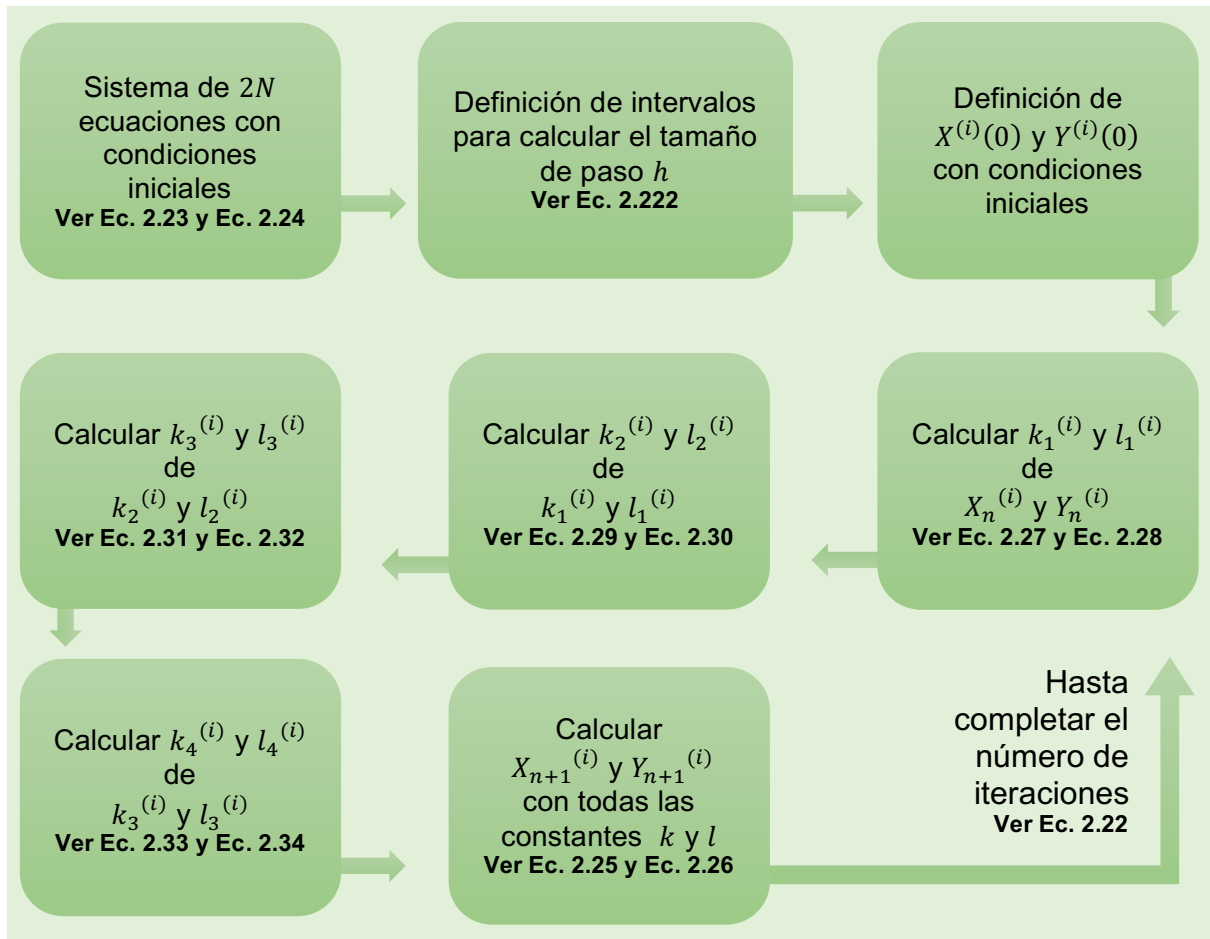


Figura 8. Proceso a seguir para utilizar Runge-Kutta de cuarto orden como método numérico para la solución de un sistema de $2N$ ecuaciones diferenciales¹⁴ (21).

- *Plan de trabajo*

En la primer y segunda semana se realizaron las actividades administrativas para garantizar el acceso a la oficina dónde trabajaríamos y al CUCEI, también se realizó el proceso para obtener internet y con esto ser más eficientes. Mientras tanto, a la par, conocimos el estado del arte analizando bibliografía en artículos científicos y libros. Además, cada viernes hasta la semana 5, el Dr. Gustavo López nos proporcionaba un curso, del cual se generaron notas en LATEX a lo largo de las semanas. Todo esto con el objetivo de obtener una capacitación necesaria para

¹⁴ El subíndice n hace referencia a la iteración n -ésima.

poder resolver los problemas y generar las simulaciones. En la quinta semana se determinó el sistema físico matemáticamente.

Una vez teniendo la matemática para desarrollar el sistema físico y la teoría necesaria para desarrollar la física de un qubit, se revisó minuciosamente el procedimiento para identificar posibles errores. Se llegó a un sistema de ecuaciones diferenciales del cual ya se tenía una solución analítica. Después de esto se elaboró un programa en PYTHON para solucionar las ecuaciones diferenciales que se obtuvieron del sistema mediante la implementación de un método numérico.

Para este primer SC surgieron algunos impedimentos al desarrollar el código ya que se hizo una investigación sobre los métodos numéricos necesarios para solucionar el sistema de ecuaciones diferenciales acopladas, pero cabía la duda de si la implementación era correcta debido a que no se estaban obteniendo los resultados esperados. Es por esta razón se solicitó una asesoría del Doctor Riemann Cruz en las instalaciones del ITESO. Esto ayudo a clarificar que en efecto el método numérico era correcto pero que había un problema en la lógica de programación y se corrigió correspondientemente.

Uno de los objetivos del proyecto es desarrollar el código en FORTRAN por lo que fue necesario traducir los resultados obtenidos en PYTHON a este otro lenguaje. Subsiguientemente, se hizo un análisis del sistema físico ahora con dos qubits. El procedimiento para resolución fue el mismo: análisis matemático, programación en PYTHON y traducción a FORTRAN. En la actualidad, se está realizando la verificación matemática para 3 qubits. Afortunadamente, no es necesario agendar reuniones ni asesorías con el Dr. Gustavo López debido a que nos encontramos trabajando en su oficina por lo que se muestran los avances de resultados y hay clarificación de dudas dependiendo de la disponibilidad del Doctor. Todas estas acciones se pueden observar de manera gráfica y resumida en la **tabla 1**. Los colores utilizados hacen referencia a tiempo en días que se utilizaron para llevar a

cabo las tareas en la semana donde rojo equivale a un día, amarillo a dos días y verde tres días.

También, se añadieron las actividades proyectadas por hacer y el tiempo esperado que se piensa que tomarán llevarlas a cabo hasta la semana 16. En la **tabla 2** se indica la simbología de los recursos utilizados para cada actividad.

Tabla 1. Cronograma de actividades de la semana 1 a la 16.

Actividades	Recursos	Semanas																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Actividades Administrativas																		
Acceso a oficina	I, LL, DG	Red																
Acceso a Internet			Red															
Capacitación																		
Lectura bibliográfica	DG, OG	Green	Green	Green	Green	Green												
Cursos de computación cuántica		Red	Red	Red	Red	Red					Red							
Determinación del Sistema físico						Red												
Documentación																		
Notas en LATEX de computación cuántica	I, C, DG, OG		Green	Green	Green	Green	Green					Green	Green	Green	Green	Green	Green	
Actividades de investigación																		
Verificación de matemática 1 qubit	I, C, DG, OG, IT						Red											
Programación en PYTHON 1 qubit							Yellow											
Programación en FORTRAN 1 qubit.									Red									
Verificación de matemática 2 qubit									Red									
Programación en PYTHON 2 qubit									Yellow									
Programación en FORTRAN 2 qubit.											Yellow							
Verificación de matemática 3 qubit												Green						
Programación en PYTHON 3 qubit												Green	Yellow					
Programación en FORTRAN 3 qubit.													Yellow	Green				
Verificación de matemática n qubit													Green	Green				
Programación en PYTHON n qubit														Green	Green			
Programación en FORTRAN n qubit.																Green	Green	

Continuación Tabla 1. Cronograma de actividades de la semana 1 a la 16

Actividades	Recursos	Semanas															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Simulación																	
Determinación de compuerta X	I, C, DG, OG																
Determinación de compuerta Y																	
Determinación de compuerta Z																	
Determinación de compuerta Hadamard																	
Determinación de compuerta CNOT																	
Determinación de compuerta CCNOT																	
Algoritmo de teletransportación cuántica																	
Algoritmo de búsqueda de Shör																	
Algoritmo de factorización de números																	
Otras Actividades																	
Asesorías	DR																

Tabla 2. Simbología de los recursos utilizados

Código	Recurso
I	Internet
LL	Llaves
DG	Dr. Gustavo López
OG	Oficina del Dr. Gustavo López
C	Computadora
DR	Dr. Riemann Cruz
IT	Instalaciones del ITESO

- *Desarrollo de propuesta de mejora*

Una vez determinado el sistema físico para un qubit, se procedió a hacer el análisis matemático teórico para revisar si el procedimiento era correcto. Para esto, se implementó la ecuación de Schrödinger descrita en la ecuación 2.21 utilizando un campo magnético \vec{B} (véase **Figura 9**) descrito en la ecuación

$$\vec{B} = (b\cos(\theta), -b\sin(\theta), B(z)); \theta = \omega t + \phi \quad [2.35]$$

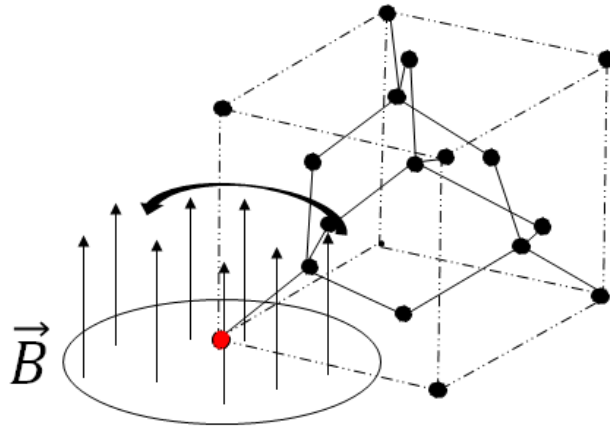


Figura 9. Sistema físico para un qubit (21).

Para llegar a una forma explícita que permite entender cómo resolver la ecuación 2.21 de manera sencilla considerando que solo se tiene un qubit, se siguieron los siguientes pasos:

1. Poner el vector Spin en términos de sus componentes en “x”, “y” y “z”.
2. Hacer el producto punto con el campo magnético.
3. Poner las funciones seno y coseno en la forma de exponencial compleja.
4. Factorizar el resultado y cambiar al operador spin de subida y bajada.
5. Definir frecuencia de Rabi (Ω) y una frecuencia de Larmor ($\tilde{\omega}$), obtenidas a partir de la multiplicación de la constante giromagnética descrita en la ecuación 2.18 y los coeficientes constantes $B(z)$ y b del campo magnético.
6. Considerar la energía de interacción de los spines, que en este caso es nula por ser el caso de un qubit.

Gracias a esto se puede poner la ecuación equivalente de la cual se puede diferenciar una parte con dependencia temporal y otra espacial, es fácil ver que todo lo que tiene dependencia en θ es la parte temporal.

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = \left(\tilde{\omega} S_z - \frac{\Omega}{2} (e^{i\theta} S_+ + e^{-i\theta} S_-) \right) |\psi\rangle \quad [2.36]$$

Por lo tanto, resolviendo la ecuación para $\Omega = 0$, se tiene un análisis que no depende del tiempo, por lo que se puede llegar a una ecuación de Eigenvalores que determina las energías correspondientes a cada estado. Las energías determinadas se pueden observar en la **Figura 10**.

$$\begin{aligned} |1\rangle &\longrightarrow E_1 = (+) \frac{\hbar\omega}{2} : \uparrow (\text{spin} - \text{arriba}) \\ |0\rangle &\longrightarrow E_0 = (-) \frac{\hbar\omega}{2} : \downarrow (\text{spin} - \text{abajo}) \end{aligned}$$

Figura 10. Energías correspondientes al estado base y al primer estado excitado. Para el estado base se tiene que el spin es antiparalelo a la dirección del campo magnético, y por lo tanto hay una energía negativa asociada al mismo. En cambio, el primer estado excitado tiene una energía positiva asociada, lo cual nos da información de que el spin está paralelo al campo magnético (21).

Posterior a esto, se hizo un análisis de la ecuación para $\Omega \neq 0$. Para esto se llevaron a cabo los siguientes pasos para llegar al resultado:

1. Se propuso la solución del sistema como una función de onda similar a la ecuación 2.6. donde los coeficientes tienen dependencia temporal.
2. Se sustituyó esta forma en la ecuación 2.36.
3. Se determinó la aplicación de los operadores en el estado propuesto, creando una ecuación en general.
4. Se aplicó una condición de normalización multiplicando toda la función por el bra $\langle 0|$ entonces debido a la ortogonalidad de los estados, lo único que queda son aquellos coeficientes que multiplican al ket $|0\rangle$. Esto generó una ecuación diferencial.

5. Se aplicó la otra condición de normalización multiplicando ahora la función por el bra $\langle 1|$ generando otra ecuación diferencial.
6. Se implementó un cambio de variable estratégico para los coeficientes con dependencia temporal dónde

$$C_n = e^{\frac{i\omega t}{2}} D_n \quad [2.37]$$

Este cambio cumple con la ecuación de Schrödinger, la normalización y además con las condiciones de frontera, también permite eliminar las frecuencias rápidas haciendo que la curva a resolver por el método numérico sea más suave y más preciso (20).

7. Posteriormente se procedió a resolver este sistema analíticamente debido a que es uno de los casos sencillos dónde se puede hacer. Aplicando la condición inicial dónde se tiene que el qubit inicial entra en un estado de 100% en el estado $|0\rangle$, se llegó a la solución de ecuaciones de los coeficientes

$$D_0(t) = e^{\frac{-i}{2}(\tilde{\omega}-\omega)t} \left(\cos \left(\frac{\sqrt{(\tilde{\omega}-\omega)^2 + \Omega^2}}{2} t \right) + \frac{i(\tilde{\omega}-\omega)}{\sqrt{(\tilde{\omega}-\omega)^2 + \Omega^2}} \sin \left(\frac{\sqrt{(\tilde{\omega}-\omega)^2 + \Omega^2}}{2} t \right) \right) \quad [2.38]$$

$$D_1(t) = \frac{i\Omega}{\sqrt{(\tilde{\omega}-\omega)^2 + \Omega^2}} e^{i(\frac{1}{2}(\tilde{\omega}-\omega)t - \phi)} \sin \left(\frac{\sqrt{(\tilde{\omega}-\omega)^2 + \Omega^2}}{2} t \right) \quad [2.39]$$

8. Para poder programarlo, no se tomó en cuenta la solución analítica y el sistema de ecuaciones obtenido se transformó a otro de cuatro debido a que se tenía un sistema de ecuaciones diferenciales ordinarias acopladas y complejas, por lo que se representó cada coeficiente dependiente del tiempo como un número complejo; y también los exponenciales imaginarios se representaron en su forma de senos y cosenos gracias a la fórmula de Euler.
9. Se hizo la multiplicación correspondiente y se igualó la parte real de una ecuación con la otra y la parte imaginaria con la parte imaginaria de la otra.

10. Finalmente se llegó a un sistema de ecuaciones diferenciales ordinarias con cuatro ecuaciones

$$\dot{X}_0 = \frac{\Omega}{2} (X_1\beta - Y_1\alpha) \quad [2.40]$$

$$\dot{Y}_0 = \frac{\Omega}{2} (X_1\alpha + Y_1\beta) \quad [2.41]$$

$$\dot{X}_1 = -\frac{\Omega}{2} (X_0\beta + Y_0\alpha) \quad [2.42]$$

$$\dot{Y}_1 = \frac{\Omega}{2} (X_0\alpha - Y_0\beta) \quad [2.43]$$

donde los coeficientes α y β se definen como

$$\alpha = \cos((\tilde{\omega} - \omega)t - \phi); \beta = \sin((\tilde{\omega} - \omega)t - \phi) \quad [2.44]$$

Una vez que se genera el sistema de ecuaciones diferenciales ordinarias, se comenzó a programar el código en PYTHON debido a su fluidez y a que tenemos más dominio en este mismo que en FORTRAN. El código en PYTHON realizado para un qubit se puede ver en el **Anexo 1**.

Las gráficas que se generaron son el módulo al cuadrado de los coeficientes para determinar la probabilidad de que el sistema se encuentre en cierto estado, en función del tiempo. Estas se pueden observar en la **Figura 11**.

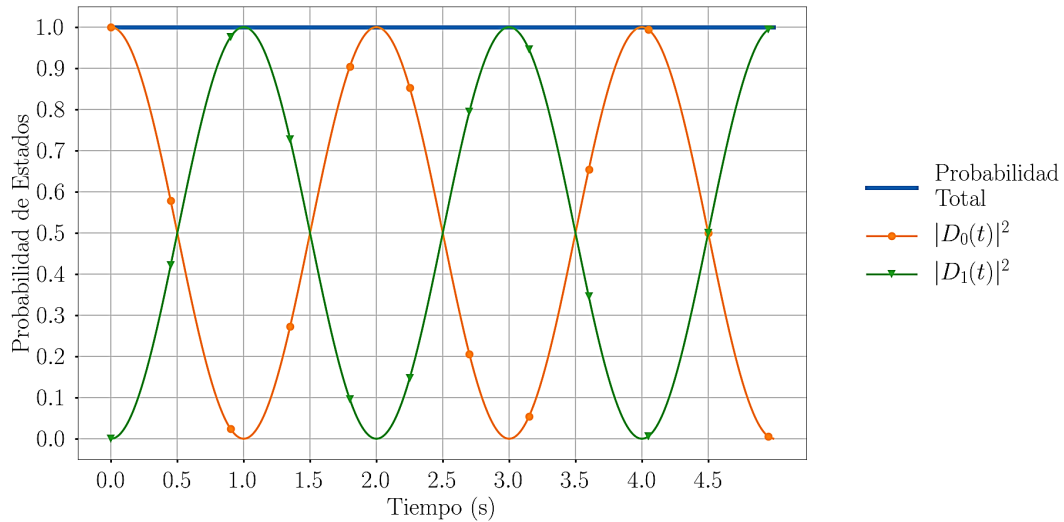


Figura 11: Rotación de un qubit (one qubit rotation gate). Esta es la compuerta asociada a la rotación de un qubit. Se puede ver una transición del estado $|0\rangle$ al $|1\rangle$ ($|0\rangle \leftrightarrow |1\rangle$). Se le dieron los valores a las frecuencias de $\omega_0 = 1/\pi, \Omega = 0.1/\omega_0, \omega = \omega_t = 100/\omega_0$ (21).

Debido a que se tenía la solución analítica, se conocía de antemano la forma de la gráfica que se debería obtener. Esta se puede ver en la **Figura 12**.

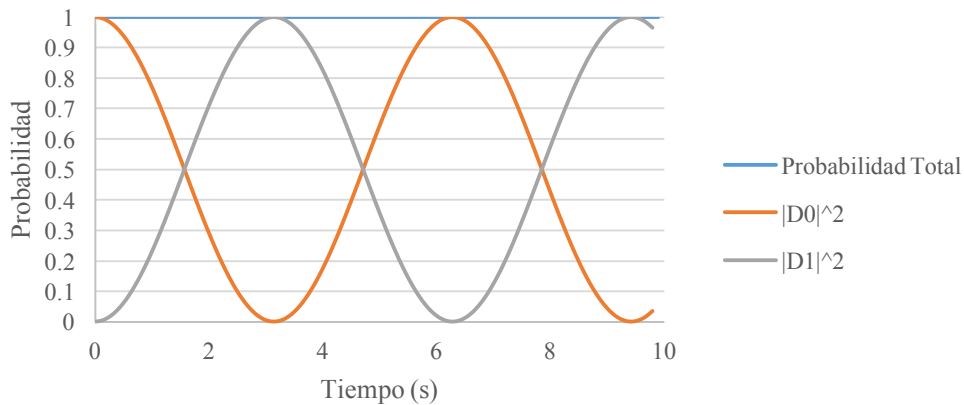


Figura 12: Solución analítica para la compuerta de rotación un qubit. Se tomaron los mismos valores de frecuencias (21).

Es importante notar que no estábamos interesando en que ambas gráficas se notaran exactamente igual, sino que el comportamiento de ambas fuera el mismo y

la probabilidad total se conservara; esto se debe a que la solución analítica había sido forzada bajo ciertas condiciones y el código general para 1 qubit cuenta con la libertad de variar parámetros

Respecto a el caso de dos qubits, se utilizan las mismas consideraciones, sin embargo, ahora hay dos energías potenciales asociadas a qubit y también existe otra energía potencial asociada a la interacción entre los qubits.

Se siguió el un procedimiento similar para llegar al resultado:

1. Primero, se usó la aproximación de Ising (20) en la cual solamente se toma en cuenta la interacción de los qubits de spin en el eje z, sin alguna contribución en el eje “x” y el eje “y”.
2. Después, se utilizó el Hamiltoniano total, dónde se tiene el mismo campo magnético utilizado en la ecuación 2.35; se expresa como

$$\hat{H} = -\vec{m}_1 \cdot \vec{B} - \vec{m}_2 \cdot \vec{B} + \frac{J}{\hbar} S_z^{(1)} S_z^{(2)} \quad [2.45]$$

El sistema cuántico con este esquema de interacción se puede observar en la **Figura 13**.

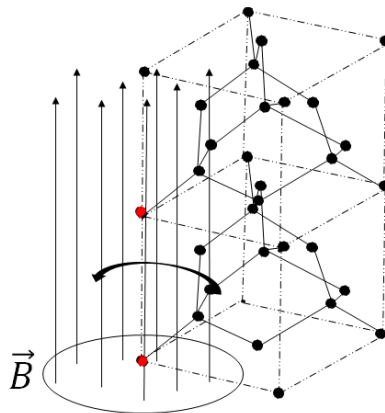


Figura 13. Sistema físico para dos qubits (21).

3. Luego, se puso la ecuación de Schrödinger de manera explícita, análogamente como en la ecuación 2.36, de ello se obtiene

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = \left(-\omega_1 S_z^{(1)} - \omega_2 S_z^{(2)} - \frac{J}{\hbar} S_z^{(1)} S_z^{(2)} - \frac{\Omega}{2} \left(e^{i\theta} S_+^{(1)} + e^{-i\theta} S_-^{(1)} + e^{i\theta} S_+^{(2)} + e^{-i\theta} S_-^{(2)} \right) \right) |\psi\rangle \quad [2.46]$$

Las frecuencias utilizadas son las mismas dónde se definió una frecuencia de Rabi (Ω) y dos frecuencias de Larmor (ω_1, ω_2).

Por lo tanto, siguiendo los mismos pasos para el qubit uno, resolviendo la ecuación para $\Omega = 0$, se tiene un análisis que no depende del tiempo. Las energías determinadas se pueden observar en la **Figura 14**.

$$\begin{aligned} |11\rangle &\longrightarrow E_{11} = -\frac{\hbar}{2} \left(-\omega_1 - \omega_2 + \frac{J}{2} \right) : \quad \uparrow \uparrow (\text{spin} - \text{arriba}, \text{arriba}) \\ |10\rangle &\longrightarrow E_{10} = -\frac{\hbar}{2} \left(\omega_1 - \omega_2 - \frac{J}{2} \right) : \quad \uparrow \downarrow (\text{spin} - \text{arriba}, \text{abajo}) \\ |01\rangle &\longrightarrow E_{01} = -\frac{\hbar}{2} \left(-\omega_1 + \omega_2 - \frac{J}{2} \right) : \quad \downarrow \uparrow (\text{spin} - \text{abajo}, \text{arriba}) \\ |00\rangle &\longrightarrow E_{00} = -\frac{\hbar}{2} \left(\omega_1 + \omega_2 + \frac{J}{2} \right) : \quad \downarrow \downarrow (\text{spin} - \text{abajo}, \text{abajo}) \end{aligned}$$

Figura 14: Niveles de energía correspondientes a los estados de dos qubits. Comparando el estado $|00\rangle$ con el $|11\rangle$ se puede ver que el primero siempre tiene una menor energía para las ω_i y J (esto se da debido a que siempre son tomadas como constantes positivas), es por esto que al estado base se le considera de menor energía; no obstante, los demás estados intermedios $|10\rangle$ y $|01\rangle$ pueden variar en energía dependiendo de la magnitud de ω_1 y ω_2 (21).

Así mismo, posterior a esto se hizo un análisis de la ecuación para $\Omega \neq 0$. Para esto se llevaron a cabo los siguientes pasos para llegar al resultado que, de igual manera, son muy similares a los que se utilizaron para el qubit 1:

1. Se propuso la solución del sistema como una función de onda similar a la ecuación 2.8 donde los coeficientes tienen dependencia temporal.
2. Se sustituyó esta forma en la ecuación 2.44.
3. Se determinó la aplicación de los operadores en el estado propuesto, creando una ecuación en general.

4. Se aplicó una condición de normalización multiplicando toda la función por el bra $\langle 00|$, $\langle 01|$, $\langle 10|$ y $\langle 11|$ individualmente generando 4 ecuaciones diferenciales acopladas.
5. Posteriormente, se implementó un cambio de variable estratégico para los coeficientes con dependencia temporal dónde

$$C_{jk} = e^{\frac{-iE_{jk}}{\hbar}t} D_{jk} \quad [2.47]$$

6. Se transformó el sistema de cuatro ecuaciones a otro de ocho debido a que eran complejas. Se hizo la multiplicación correspondiente y se igualó la parte real de una ecuación con la otra y la parte imaginaria con la parte imaginaria de la otra.
7. Finalmente se llegó a un sistema de ecuaciones diferenciales ordinarias con ocho ecuaciones

$$\dot{X}_{00} = \frac{\Omega}{2} (-X_{01} \sin(\alpha_{01}^{00}) - Y_{01} \cos(\alpha_{01}^{00}) - X_{10} \sin(\alpha_{10}^{00}) - Y_{10} \cos(\alpha_{10}^{00})) \quad [2.48]$$

$$\dot{Y}_{00} = \frac{\Omega}{2} (X_{01} \cos(\alpha_{01}^{00}) - Y_{01} \sin(\alpha_{01}^{00}) + X_{10} \cos(\alpha_{10}^{00}) - Y_{10} \sin(\alpha_{10}^{00})) \quad [2.49]$$

$$\dot{X}_{01} = \frac{\Omega}{2} (X_{00} \sin(\alpha_{01}^{00}) - Y_{00} \cos(\alpha_{01}^{00}) + X_{11} \sin(\alpha_{01}^{11}) - Y_{11} \cos(\alpha_{01}^{11})) \quad [2.50]$$

$$\dot{Y}_{01} = \frac{\Omega}{2} (X_{00} \cos(\alpha_{01}^{00}) + Y_{00} \sin(\alpha_{01}^{00}) + X_{11} \cos(\alpha_{01}^{11}) + Y_{11} \sin(\alpha_{01}^{11})) \quad [2.51]$$

$$\dot{X}_{10} = \frac{\Omega}{2} (X_{00} \sin(\alpha_{10}^{00}) - Y_{00} \cos(\alpha_{10}^{00}) + X_{11} \sin(\alpha_{10}^{11}) - Y_{11} \cos(\alpha_{10}^{11})) \quad [2.52]$$

$$\dot{Y}_{10} = \frac{\Omega}{2} (X_{00} \cos(\alpha_{10}^{00}) + Y_{00} \sin(\alpha_{10}^{00}) + X_{11} \cos(\alpha_{10}^{11}) + Y_{11} \sin(\alpha_{10}^{11})) \quad [2.53]$$

$$\dot{X}_{11} = \frac{\Omega}{2} (-X_{01} \sin(\alpha_{01}^{11}) - Y_{01} \cos(\alpha_{01}^{11}) - X_{10} \sin(\alpha_{10}^{11}) - Y_{10} \cos(\alpha_{10}^{11})) \quad [2.54]$$

$$\dot{Y}_{11} = \frac{\Omega}{2} (X_{01} \cos(\alpha_{01}^{11}) - Y_{01} \sin(\alpha_{01}^{11}) + X_{10} \cos(\alpha_{10}^{11}) - Y_{10} \sin(\alpha_{10}^{11})) \quad [2.55]$$

Una vez que se obtuvo el sistema de ecuaciones diferenciales ordinarias, se comenzó a programar el código en PYTHON, y posteriormente en FORTRAN. Programar esto fue más sencillo debido a la similitud que se tenía con el de un qubit. El código en PYTHON realizado se puede ver en el **Anexo 2**.

Se hicieron las mismas consideraciones para la interpretación de resultados que con un qubit. Los resultados se pueden observar en la **Figura 15**, dónde se muestra la evolución temporal de las probabilidades de los estados.

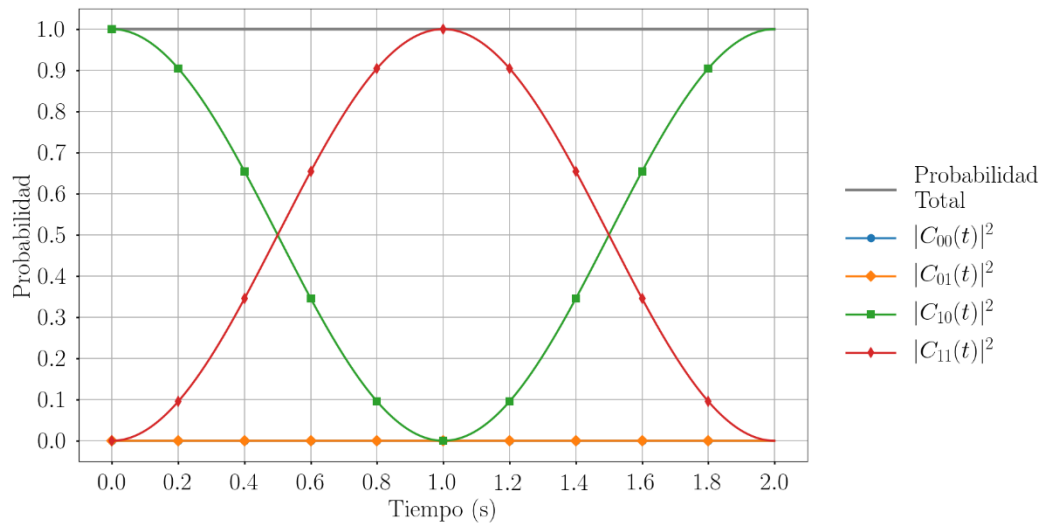


Figura 15: Transición de probabilidades de estados para dos qubits. Hay transiciones de probabilidad de estados donde el ket $|ii\rangle$ tiene una probabilidad $|C_{ii}|^2$. Se le dio valores a $\omega_0 = 1/\pi, \Omega = 1/\omega_0, \omega_1 = 100/\omega_0, \omega_2 = 200/\omega_0, J = 0$. Con un estado $|10\rangle$ inicial (21).

Así mismo, modificando un poco los parámetros iniciales, para simular una compuerta CNOT a un tiempo dado. Se le dio valores a $\Omega = 0.1, \omega = 100, \omega_1 = 100, \omega_2 = 0$ con un qubit inicial $|10\rangle$. Esto se puede ver en la **Figura 16**.

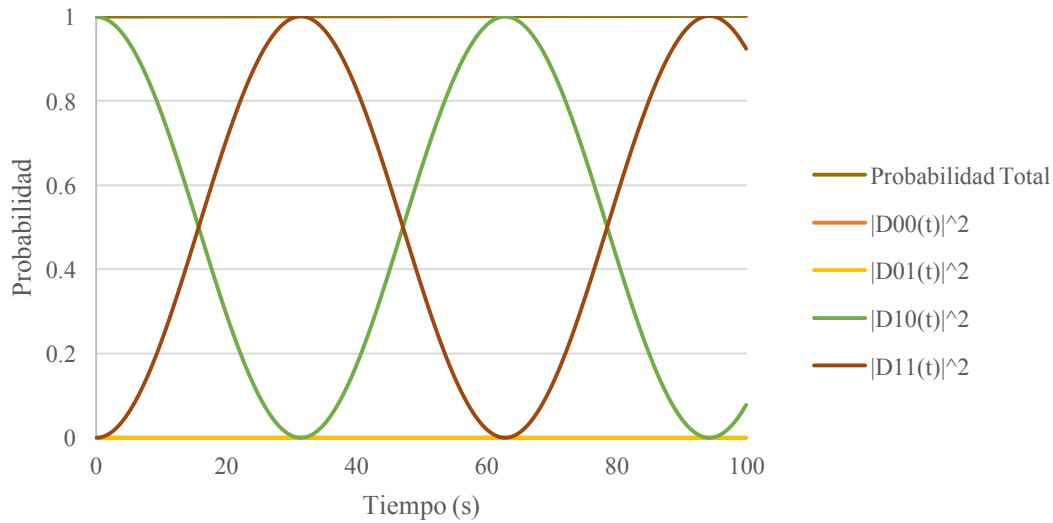


Figura 16: Transición de probabilidades de estados para dos qubits implementando una compuerta CNOT. La comparación del tiempo inicial y el tiempo de $\pi/\Omega \approx 31.41$ segundos, se puede observar una transición del estado $|10\rangle$ al estado $|11\rangle$, siendo lo esperado en la literatura (20). El estado $|00\rangle$ y $|01\rangle$ se mantienen con probabilidad 0 durante todo el tiempo (21).

Una vez que se logró realizar el código para 1 y 2 qubits en PYTHON, se desarrollaron los códigos para 3 qubits en ambos lenguajes de programación. Después de esto se comenzó a ver pequeños detalles que había en cada código, así como un programa en general que graficara los datos obtenidos a partir de FORTRAN.

Para el análisis matemático, se siguió el un procedimiento similar para llegar al sistema de ecuaciones:

1. Como en el caso de dos qubits, se usó la aproximación de Ising (20), posteriormente se determinó la energía de interacción entre los qubits a primer y segundos vecinos.
2. Después, se utilizó el Hamiltoniano total, dónde se tiene el mismo campo magnético. El sistema físico se puede observar en la **Figura 17**.

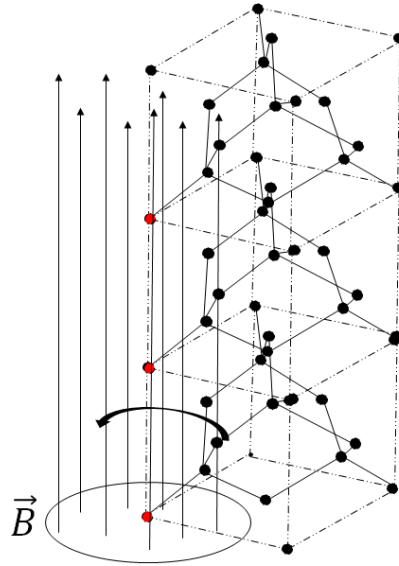


Figura 17. Sistema físico para tres qubits (21).

3. Luego, se obtuvo la ecuación de Schrödinger correspondiente. Por facilidad se realizó el cambio de los kets observado en la **Figura 18**.

$$i\hbar \frac{\partial}{\partial t} |\Psi\rangle = (\hat{H}_o + \hat{\omega}(t)) |\Psi\rangle \quad [2.56]$$

$ 111\rangle$	\longrightarrow	$ 8\rangle$
$ 110\rangle$	\longrightarrow	$ 7\rangle$
$ 101\rangle$	\longrightarrow	$ 6\rangle$
$ 100\rangle$	\longrightarrow	$ 5\rangle$
$ 011\rangle$	\longrightarrow	$ 4\rangle$
$ 010\rangle$	\longrightarrow	$ 3\rangle$
$ 001\rangle$	\longrightarrow	$ 2\rangle$
$ 000\rangle$	\longrightarrow	$ 1\rangle$

Figura 18. Cambio de notación para los estados de tres qubits (21).

Donde

$$|\Psi\rangle = \sum_{i=1}^8 C_i(t) |i\rangle \quad [2.57]$$

$$\hat{H}_o = - \sum_{i=1}^3 \tilde{\omega}_i S_z^{(i)} - \frac{1}{\hbar} \left[J \left(\vec{S}_1 \cdot \vec{S}_2 + \vec{S}_1 \cdot \vec{S}_3 \right) + J' \vec{S}_1 \cdot \vec{S}_3 \right] \quad [2.58]$$

$$\hat{\omega}(t) = -\frac{\Omega}{2} \sum_{i=1}^3 \left(e^{i\theta} S_+^{(i)} + e^{-i\theta} S_-^{(i)} \right) \quad [2.59]$$

4. Se resolvió para $\Omega = 0$, obteniendo una expresión para las energías de manera general

$$\frac{1}{\hbar} E_{\alpha_3 \alpha_2 \alpha_1} = -\frac{1}{2} \left[\sum_{i=1}^3 \tilde{\omega}_i (-1)^{\alpha_i} + \frac{J}{2} \left((-1)^{\alpha_1 + \alpha_2} + (-1)^{\alpha_2 + \alpha_3} \right) + \frac{J'}{2} (-1)^{\alpha_1 + \alpha_3} \right] \quad [2.60]$$

Posterior a esto se hizo un análisis de la ecuación de Schrödinger para $\Omega \neq 0$. Se propuso la solución del sistema como una función de onda similar a la ecuación 2.8 pero más general para 3 qubits donde los coeficientes tienen dependencia temporal.

1. El cambio de notación de la **Figura 18** se sustituyó en la ecuación de manera general.
2. Se aplicó una condición de normalización multiplicando toda la función por el bra $\langle k|$ donde $k = 1, 2, \dots, 8$ respectivamente, con esto se puede obtener una matriz para el operador $[\tilde{\omega}(t)]$, esta se puede observar en la ecuación 2.61.

$$[\tilde{\omega}(t)] = \begin{pmatrix} 0 & z & z & 0 & z & 0 & 0 & 0 \\ z^* & 0 & 0 & z & 0 & z & 0 & 0 \\ z^* & 0 & 0 & z & 0 & 0 & z & 0 \\ 0 & z^* & z^* & 0 & 0 & 0 & 0 & z \\ z^* & 0 & 0 & 0 & 0 & z & z & 0 \\ 0 & z^* & 0 & 0 & z^* & 0 & 0 & z \\ 0 & 0 & z^* & 0 & z^* & 0 & 0 & z \\ 0 & 0 & 0 & z^* & 0 & z^* & z^* & 0 \end{pmatrix} \quad [2.61]$$

Donde $z(t) = e^{i\theta(t)}$

3. Esto a su vez genera ecuaciones diferenciales acopladas de las cuales se implementó un cambio de variable estratégico para los coeficientes con dependencia temporal similar al de las ecuaciones 2.37 y 2.47.
4. Se transformó el sistema de ocho ecuaciones a otro de dieciséis debido a que eran complejas. Se hizo la multiplicación correspondiente y se igualó la parte real de una ecuación con la otra y la parte imaginaria con la parte imaginaria de la otra.
5. Finalmente se llegó a un sistema de ecuaciones diferenciales ordinarias con dieciséis ecuaciones.

$$\dot{X}_1 = \frac{U}{2} [-X_2 \sin(\beta_{12}) - Y_2 \cos(\beta_{12}) - X_3 \sin(\beta_{13}) - Y_3 \cos(\beta_{13}) - X_5 \sin(\beta_{15}) - Y_5 \cos(\beta_{15})] \quad [2.62]$$

$$\dot{Y}_1 = \frac{U}{2} [X_2 \cos(\beta_{12}) - Y_2 \sin(\beta_{12}) + X_3 \cos(\beta_{13}) - Y_3 \sin(\beta_{13}) + X_5 \cos(\beta_{15}) - Y_5 \sin(\beta_{15})] \quad [2.63]$$

$$\dot{X}_2 = \frac{U}{2} [X_1 \sin(\beta_{12}) - Y_1 \cos(\beta_{12}) - X_4 \sin(\beta_{24}) - Y_4 \cos(\beta_{24}) - X_6 \sin(\beta_{26}) - Y_6 \cos(\beta_{26})] \quad [2.64]$$

$$\dot{Y}_2 = \frac{U}{2} [X_1 \cos(\beta_{12}) + Y_1 \sin(\beta_{12}) + X_4 \cos(\beta_{24}) - Y_4 \sin(\beta_{24}) + X_6 \cos(\beta_{26}) - Y_6 \sin(\beta_{26})] \quad [2.65]$$

$$\dot{X}_3 = \frac{U}{2} [X_1 \sin(\beta_{13}) - Y_1 \cos(\beta_{13}) - X_4 \sin(\beta_{34}) - Y_4 \cos(\beta_{34}) - X_7 \sin(\beta_{37}) - Y_7 \cos(\beta_{37})] \quad [2.66]$$

$$\dot{Y}_3 = \frac{U}{2} [X_1 \cos(\beta_{13}) + Y_1 \sin(\beta_{13}) + X_4 \cos(\beta_{34}) - Y_4 \sin(\beta_{34}) + X_7 \cos(\beta_{37}) - Y_7 \sin(\beta_{37})] \quad [2.67]$$

$$\dot{X}_4 = \frac{U}{2} [X_2 \sin(\beta_{24}) - Y_2 \cos(\beta_{24}) + X_3 \sin(\beta_{34}) - Y_3 \cos(\beta_{34}) - X_8 \sin(\beta_{48}) - Y_8 \cos(\beta_{48})] \quad [2.68]$$

$$\dot{Y}_4 = \frac{U}{2} [X_2 \cos(\beta_{24}) + Y_2 \sin(\beta_{24}) + X_3 \cos(\beta_{34}) + Y_3 \sin(\beta_{34}) + X_8 \cos(\beta_{48}) - Y_8 \sin(\beta_{48})] \quad [2.69]$$

$$\dot{X}_5 = \frac{U}{2} [X_1 \sin(\beta_{15}) - Y_1 \cos(\beta_{15}) - X_6 \sin(\beta_{56}) - Y_6 \cos(\beta_{56}) - X_7 \sin(\beta_{57}) - Y_7 \cos(\beta_{57})] \quad [2.70]$$

$$\dot{Y}_5 = \frac{U}{2} [X_1 \cos(\beta_{15}) + Y_1 \sin(\beta_{15}) + X_6 \cos(\beta_{56}) - Y_6 \sin(\beta_{56}) + X_7 \cos(\beta_{57}) - Y_7 \sin(\beta_{57})] \quad [2.71]$$

$$\dot{X}_6 = \frac{U}{2} [X_2 \sin(\beta_{26}) - Y_2 \cos(\beta_{26}) + X_5 \sin(\beta_{56}) - Y_5 \cos(\beta_{56}) - X_8 \sin(\beta_{68}) - Y_8 \cos(\beta_{68})] \quad [2.72]$$

$$\dot{Y}_6 = \frac{U}{2} [X_2 \cos(\beta_{26}) + Y_2 \sin(\beta_{26}) + X_5 \cos(\beta_{56}) + Y_5 \sin(\beta_{56}) + X_8 \cos(\beta_{68}) - Y_8 \sin(\beta_{68})] \quad [2.73]$$

$$\dot{X}_7 = \frac{U}{2} [X_3 \sin(\beta_{37}) - Y_3 \cos(\beta_{37}) + X_5 \sin(\beta_{57}) - Y_5 \cos(\beta_{57}) - X_8 \sin(\beta_{78}) - Y_8 \cos(\beta_{78})] \quad [2.74]$$

$$\dot{Y}_7 = \frac{U}{2} [X_3 \cos(\beta_{37}) + Y_3 \sin(\beta_{37}) + X_5 \cos(\beta_{57}) + Y_5 \sin(\beta_{57}) + X_8 \cos(\beta_{78}) - Y_8 \sin(\beta_{78})] \quad [2.75]$$

$$\dot{X}_8 = \frac{U}{2} [X_4 \sin(\beta_{48}) - Y_4 \cos(\beta_{48}) + X_6 \sin(\beta_{68}) - Y_6 \cos(\beta_{68}) + X_7 \sin(\beta_{78}) - Y_7 \cos(\beta_{78})] \quad [2.76]$$

$$\dot{Y}_8 = \frac{U}{2} [X_4 \cos(\beta_{48}) + Y_4 \sin(\beta_{48}) + X_6 \cos(\beta_{68}) + Y_6 \sin(\beta_{68}) + X_7 \cos(\beta_{78}) + Y_7 \sin(\beta_{78})] \quad [2.77]$$

donde

$$\beta_{ij} = \frac{1}{\hbar} (E_i - E_j) + \theta \quad [2.78]$$

Es importante notar que existen interacciones específicas denotadas por el potencial que genera este mismo sistema de ecuaciones, esto se puede ver también en la relación existente con los coeficientes de la ecuación 2.61. Por lo que la interacción de estados se ve limitada y no puede haber transiciones de uno a otro tan libremente. Esto se puede observar de manera gráfica en la **Figura 19**.

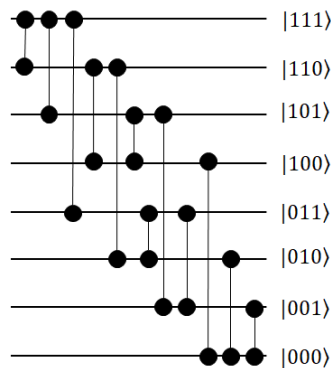


Figura 19. Reglas de selección para tres qubits. Estas relaciones se obtienen debido al potencial en el que se somete el sistema cuántico. Se puede ver que un estado interactúa con otros tres estados, por lo que una transición con otro estado diferente que no esté ligado no puede llevarse a cabo bajo la frecuencia de resonancia correspondiente para el salto de estados (21).

Una vez que se obtuvo el sistema de ecuaciones diferenciales ordinarias, se comenzó a programar el código en PYTHON, se puede observar en el **Anexo 3**.

Debido a que se obtuvo la relación que existe entre los estados debido al potencial para tres qubits, también fue necesario trabajar en las relaciones para el programa de uno y dos qubits, esas se pueden observar en la **Figura 20** y **Figura 21**, respectivamente.

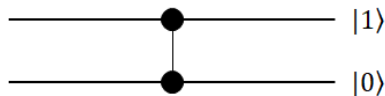


Figura 20. Reglas de selección de un qubit. Se puede apreciar la relación entre estados debido al potencial en el que se somete el sistema cuántico (21).

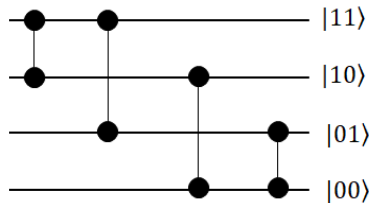


Figura 21. Reglas de selección de dos qubits. Es importante notar puede que para este caso, un estado interactúa con otros dos estados (21).

Es importante notar que cada estado interactúa con el estado que tiene un qubit diferente, en términos generales (tomando como ejemplo el caso de tres qubits), el estado $|i j k\rangle$, interactúa con los estados $|\bar{i} j k\rangle$, $|i \bar{j} k\rangle$, y $|i j \bar{k}\rangle$, donde i, j y k pueden ser 0 o 1, y el signo testado significa el opuesto (es decir si $i = 1$, entonces $\bar{i} = 0$).

Posterior a esto, lo que se realizó fue determinar las compuertas necesarias en el programa de 3 qubits para generar algoritmos, como el de teleportación cuántica.

3. Resultados del trabajo profesional

Durante las 16 semanas del semestre, fue posible realizar 3 códigos en lenguaje FORTRAN 95 y PYTHON 3.6 que dan soluciones numéricas de las ecuaciones que genera un sistema cuántico de 1, 2, y 3 qubits con spin $\frac{1}{2}$. Al mismo tiempo, se hizo un código en PYTHON 3.6 que grafica las soluciones numéricas de los códigos generados por FORTRAN.

Los códigos generales para resolver las ecuaciones de un qubit de PYTHON se encuentran en **Anexo 1**; para dos qubits, en el **Anexo 2** y para tres qubits en el **Anexo 3**. El **Anexo 4** contiene el código para graficar los resultados obtenidos en FORTRAN. Así mismo gracias a esto, se pudieron hacer los protocolos necesarios para generar los algoritmos de *Estados de Bell*, la compuerta *CCNOT* y *Teleportación Cuántica*.

Para generar compuertas en el programa, es necesario modificar la frecuencia del campo magnético haciéndola resonante con la diferencia de energías de los estados que se quieren manipular, es decir $E_i - E_j$, para hacer una transición del estado i al j (o viceversa) siempre y cuando la transición sea válida de acuerdo al marco de interacción del potencial (observar las **Figuras 19, 20 y 21**) y así mismo con la condición de que para todo caso $E_i > E_j$. De lo contrario, no habrá interacción entre los estados.

Para crear una compuerta de superposición de estados, el tiempo que se debe dejar el pulso magnético corresponde a un tiempo $\frac{\pi}{2\Omega}$ (llamado $\frac{\pi}{2}$ - pulso), en cambio para hacer el intercambio completo de estos estados se requiere un tiempo $\frac{\pi}{\Omega}$ (llamado π - pulso).

Estados de Bell (entrelazamiento cuántico)

Los Estados de Bell se pueden hacer mediante un sistema de dos o más qubits. Para hacer un Estado de Bell puro, es necesario tener dos qubits puros (ya sean dos estados $|0\rangle$, $|1\rangle$ o la combinación de estos). Posteriormente, se le aplica una compuerta Hadamard a un qubit y luego se aplica una compuerta CNOT (para ver el algoritmo, referirse a la **Figura 4**). Es importante recordar que el aplicar una compuerta Hadamard genera la superposición de estados, y después el aplicar una compuerta CNOT los correlacióna generando así el entrelazamiento cuántico.

Se generó un programa en PYTHON (referirse al **Anexo 5**) y FORTRAN, donde se crean los cuatro Estados de Bell a partir de las condiciones iniciales específicas, estos se pueden observar en las **Figuras** de la **22** a la **25**.

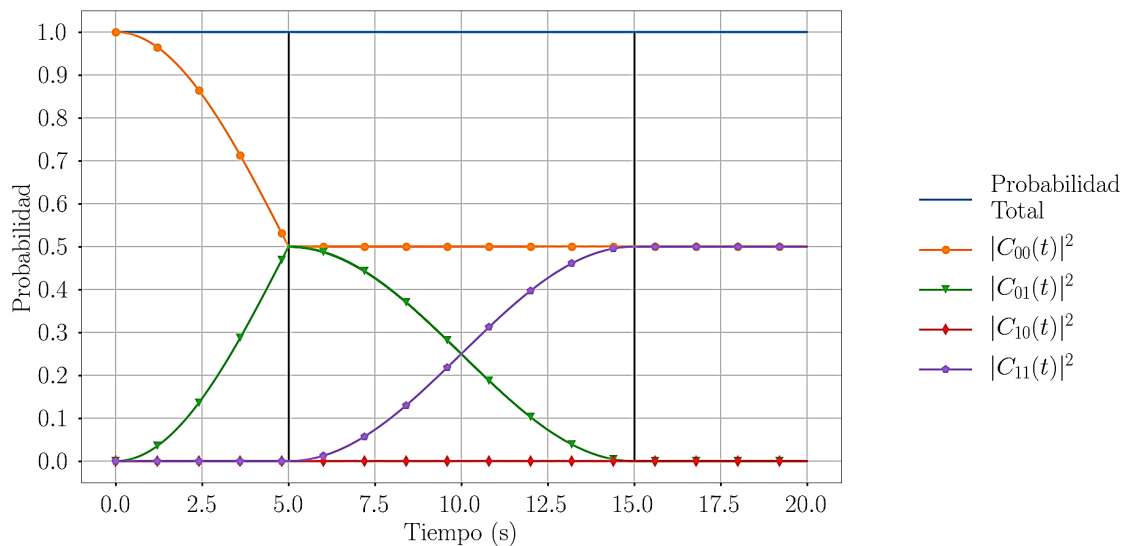


Figura 22. Estado de Bell $|\beta_{00}\rangle$ generado a partir de introducir un registro de dos qubits $|00\rangle$ (21).

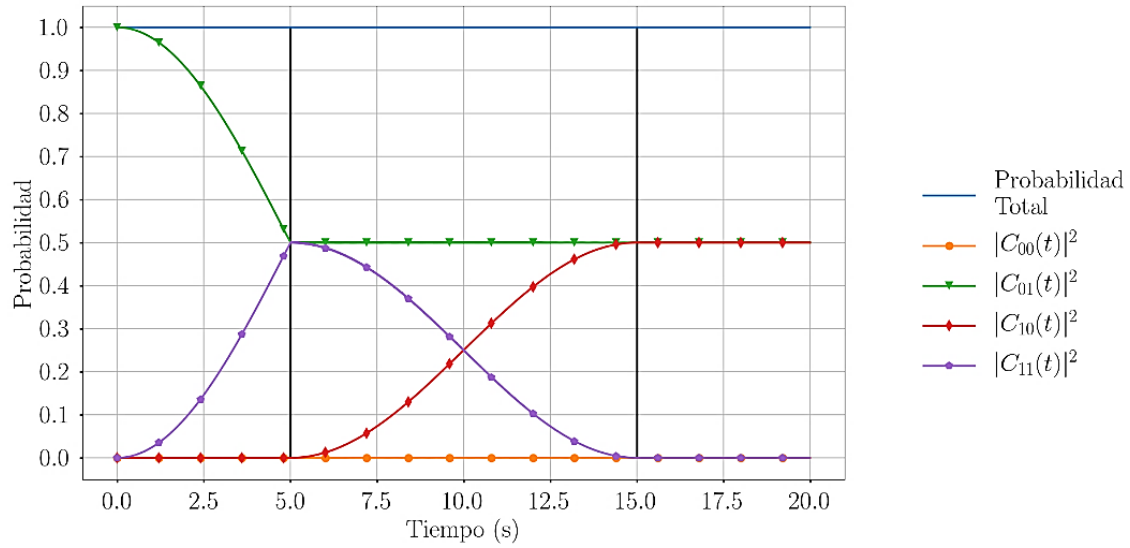


Figura 23. Estado de Bell $|\beta_{01}\rangle$ generado a partir de introducir un registro de dos qubits $|01\rangle$ (21).

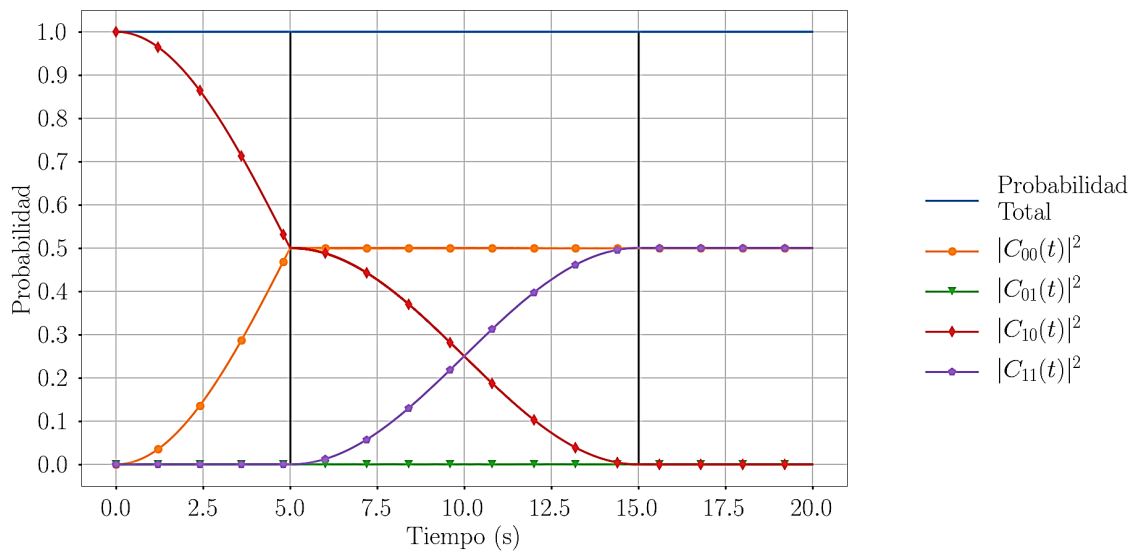


Figura 24. Estado de Bell $|\beta_{10}\rangle$ generado a partir de introducir un registro de dos qubits $|10\rangle$ (21).

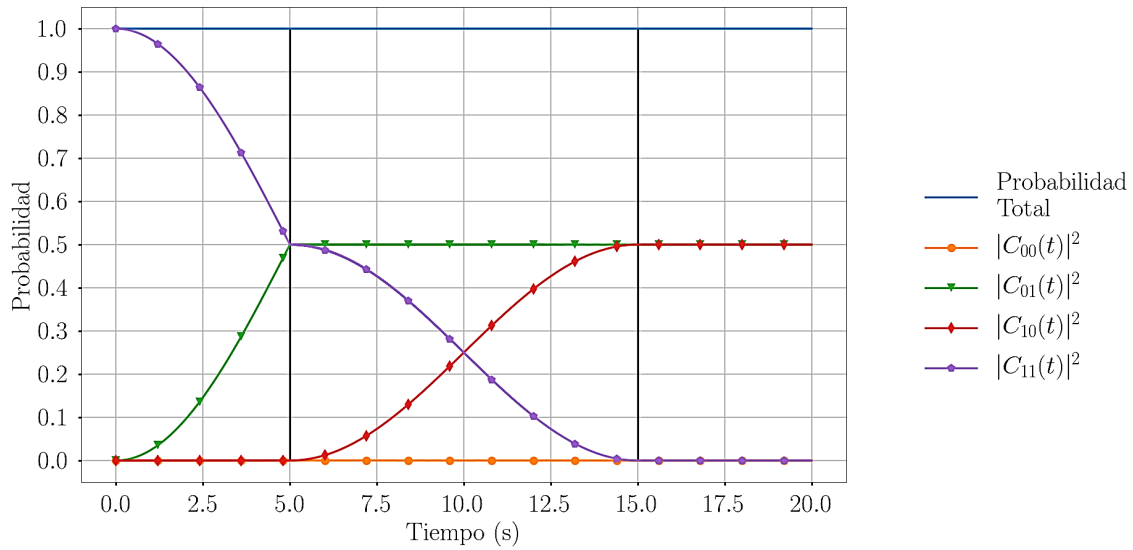


Figura 25. Estado de Bell $|\beta_{11}\rangle$ generado a partir de introducir un registro de dos qubits $|11\rangle$ (21).

Compuerta CCNOT

Otra compuerta importante para obtener es la compuerta CCNOT, (también conocida como Tofoli), esta toma en cuenta los dos primeros registros de qubits, en caso de que sean ambos el estado $|1\rangle$, aplica la negación del último estado, en caso contrario no afecta al sistema. Por lo tanto, CCNOT pasa del estado $|110\rangle$ al $|111\rangle$ o viceversa. La tabla de verdad de esta compuerta se puede observar en la **Figura 26**.

CCN					
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Figura 26. Tabla de verdad de la compuerta CCNOT (24).

Esta compuerta se puede hacer fácilmente mediante un π – pulso con una frecuencia equivalente a la diferencia de energías $E_8 - E_7$, y el resultado del programa se puede observar en la **Figura 27**.

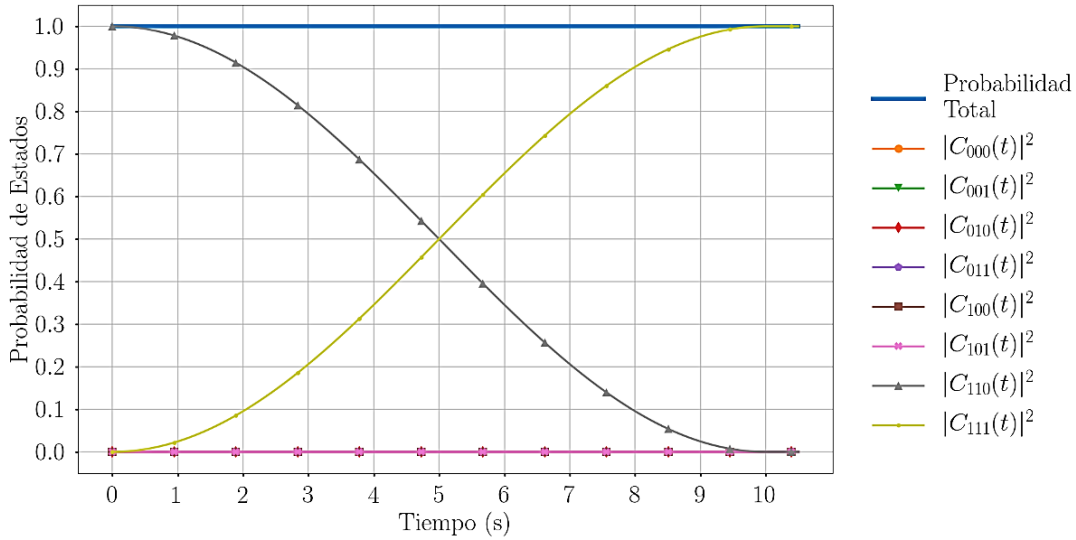


Figura 27. Compuerta CCNOT. En el programa de tres qubits, esta compuerta es un cambio entre el estado $|8\rangle$ y $|7\rangle$ (21).

En cambio, si se aplica una frecuencia distinta a la diferencia de energías, existe una transición no resonante, haciendo que el resultado se vea alterado. Esto se puede ver en la **Figura 28**.

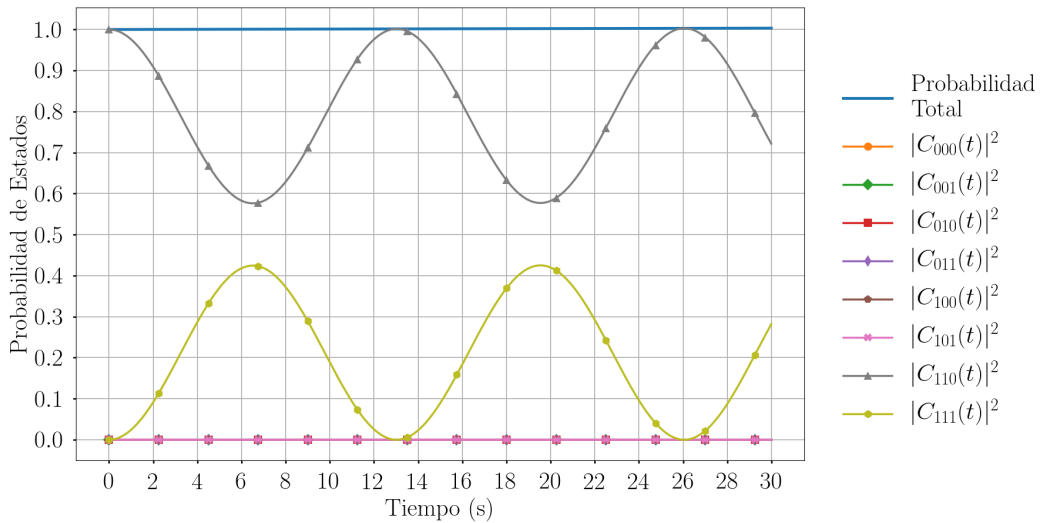


Figura 28. Transición de energías con una frecuencia no resonante, (en este caso del 99.88% de la diferencia $E_8 - E_7$) (21).

Teleportación cuántica

Para la Teleportación Cuántica, es bien conocido que se debe implementar el algoritmo de la **Figura 29**. Este algoritmo permite pasar información contenida en tercer qubit al primero sin que estos dos hayan interactuado. Para lograr este fenómeno, se necesita comenzar con un estado de Bell entre los primeros dos qubits, posteriormente, al tercer y segundo qubits se entrelazan con una compuerta CNOT y luego se aplica una compuerta Hadamard al tercero. Con este procedimiento sencillo se puede obtener la información necesaria para recuperar el estado del tercer qubit. Se pueden analizar con detenimiento las ecuaciones 3.1 a 3.5 que describen la función de onda a lo largo de cada paso del algoritmo.

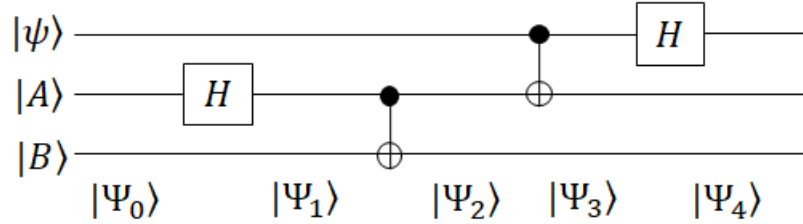


Figura 29. Algoritmo de Teleportación cuántica para un estado arbitrario $|\psi\rangle$ (21).

$$|\Psi_0\rangle = \alpha |000\rangle + \beta |100\rangle \quad [3.1]$$

$$|\Psi_1\rangle = \frac{\alpha}{\sqrt{2}} |000\rangle + \frac{\alpha}{\sqrt{2}} |010\rangle + \frac{\beta}{\sqrt{2}} |100\rangle + \frac{\beta}{\sqrt{2}} |110\rangle \quad [3.2]$$

$$|\Psi_2\rangle = \frac{\alpha}{\sqrt{2}} |000\rangle + \frac{\alpha}{\sqrt{2}} |011\rangle + \frac{\beta}{\sqrt{2}} |100\rangle + \frac{\beta}{\sqrt{2}} |111\rangle \quad [3.3]$$

$$|\Psi_3\rangle = \frac{\alpha}{\sqrt{2}} |000\rangle + \frac{\alpha}{\sqrt{2}} |011\rangle + \frac{\beta}{\sqrt{2}} |110\rangle + \frac{\beta}{\sqrt{2}} |101\rangle \quad [3.4]$$

$$|\Psi_4\rangle = \frac{\alpha}{2} |000\rangle + \frac{\alpha}{2} |100\rangle + \frac{\alpha}{2} |011\rangle + \frac{\alpha}{2} |111\rangle - \frac{\beta}{2} |110\rangle + \frac{\beta}{2} |010\rangle - \frac{\beta}{2} |101\rangle + \frac{\beta}{2} |001\rangle \quad [3.5]$$

Reorganizando la ecuación 3.5 para poder observar más fácilmente la distribución del último qubit,

$$|\Psi_4\rangle = \frac{1}{2} |00\rangle (\alpha |0\rangle + \beta |1\rangle) + \frac{1}{2} |01\rangle (\beta |0\rangle + \alpha |1\rangle) + \frac{1}{2} |10\rangle (\alpha |0\rangle - \beta |1\rangle) + \frac{1}{2} |11\rangle (-\beta |0\rangle + \alpha |1\rangle) \quad [3.6]$$

Posterior a esto, dependiendo de la observación de los dos primeros qubits se deben implementar las compuertas necesarias para restaurar el estado inicial y por ende completar la teleportación.

En el programa, se aplicaron una serie de diversos pulsos magnéticos haciendo las transiciones equivalentes para poder llegar a obtener los estados deseados en cada etapa. Hubo un problema al intentar aplicar la segunda compuerta CNOT para llegar al estado $|\psi_3\rangle$, esto se dió debido a que la diferencia de energía $E_7 - E_5$ es equivalente a la diferencia de energía $E_4 - E_2$, haciendo que hubiera una doble transición no deseada de los estados, esto representó un gran problema para intentar hacer el algoritmo paso a paso por lo que se buscó una representación equivalente del estado. En la **Figura 30** se puede apreciar la evolución de los estados del algoritmo de teleportación cuántica a partir de las simulaciones generadas en el programa de PYTHON (referirse al **Anexo 6**).

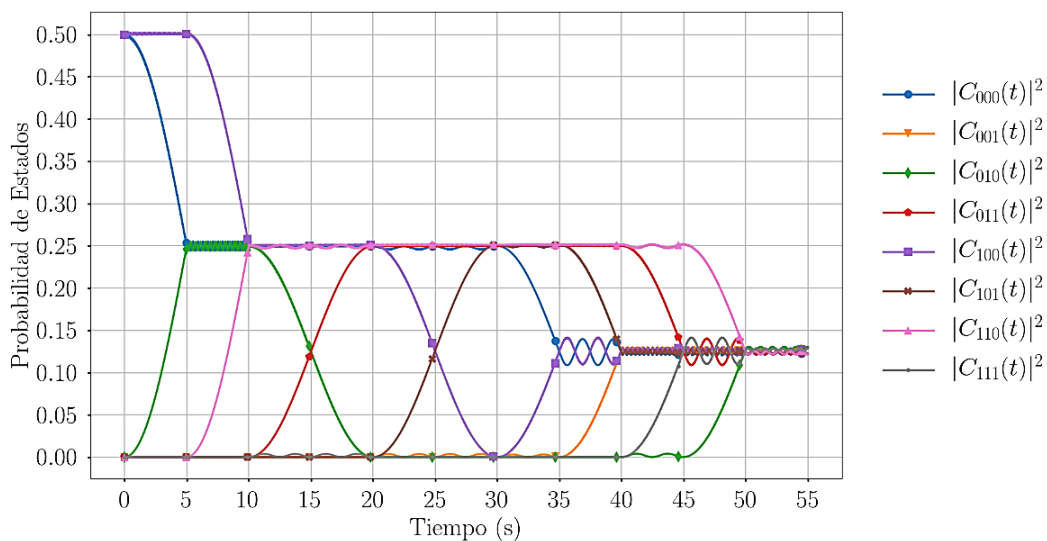


Figura 30. Probabilidad de estados en teleportación cuántica (21).

Después del último pulso, se obligó a que la frecuencia del campo magnético fuese cero para poder apreciar la relación entre los 8 estados de los 3 qubits (en el segundo 50 al segundo 55). Es importante notar que hay presencia de oscilaciones periódicas en ciertos intervalos de tiempos y, en especial, cuando la frecuencia del campo magnético es cero; esto se debe a que por la naturaleza de la interacción entre qubits, se dan oscilaciones no resonantes.

4. Reflexiones del alumno o alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto

- Aprendizajes profesionales

Daniel Hernández Mota

El llevar a cabo este proyecto me ayudó a desarrollar una serie de competencias que mejoraron mi calidad educativa y profesional. Algunas de estas competencias fueron: aprendizaje autónomo, para poder resolver algunos desafíos que se aparecían al intentar programar ya sea en PYTHON o en FORTRAN; comunicación eficaz, para poder mantener un dialogo abierto, profesional y formal informándonos acerca de los puntos importantes de la investigación; y trabajo en equipo, al paralelizar acciones para poder llevar a cabo de una manera más rápida el proyecto de investigación y también explicándonos ciertos procedimientos que no eran tan fáciles de entender, explotando de manera eficiente las habilidades de mi compañera de trabajo así como las mías, se desarrolló esta competencia trabajando de una manera colaborativa.

Profesionalmente, desarrollé aún más el pensamiento matemático utilizando herramientas que obtuve a lo largo de la carrera (como métodos numéricos, variable compleja, ecuaciones diferenciales, nanofísica I y II, cálculo tensorial, álgebra lineal) para poder llegar a las ecuaciones necesarias y solucionarlas correctamente, prácticamente pareció como si todos estos cursos que llevé por separado en la

carrera se juntaron en un mismo proyecto dándome la oportunidad de ponerlos en práctica.

En concreto, pude aprender que la inversión que se tiene en el área de investigación en México es reducida y limitada, esto a su vez, limita el área de investigación experimental, dándole una mayor importancia a la faceta teórica debido a que, en este contexto específico, llegar a desarrollar algún prototipo experimental tomaría mucho gasto desde el punto de vista económico y energético. Por tanto, tener simulaciones del mismo sistema para determinar el comportamiento es un paso crítico para determinar si es viable la experimentación. Gracias a la investigación teórica se pueden redireccionar los recursos de una manera más considerada y con mejores oportunidades de tener éxito.

Respecto al aprendizaje de mi proyecto de vida profesional, pude darme cuenta de que la investigación en física teórica es un área en la cuál podría dedicarme sin ningún problema debido a que no sólo se hace una contribución a la ciencia, sino que también esta misma puede tener un impacto más allá de lo teórico, se puede intentar buscar un área de aplicación que tenga un impacto más directo hacia la sociedad y todo bajo la misma premisa de resolver la parte teórica y posteriormente simular.

Karina Chiñas Fuentes

En este periodo de investigación, las habilidades más importantes que desarrollé fueron el análisis lógico-matemático que tuve para resolver diversos sistemas planteados desde el punto de vista de la Mecánica Cuántica, así como las diversas consideraciones que los físicos teóricos tienen como opción para abordar los problemas (tales como la aproximación de Ising y la de Born para interacción entre partículas con spin semi-entero). Además de ello, aprendí a programar en el lenguaje FORTRAN 95 y reforcé mis conocimientos en el lenguaje de PYTHON; así como aprender a usar mejor mi sistema operativo en *Linux*, ya que ahí era donde

se generaban, probaban y corrían todos los códigos para asegurar su buen funcionamiento.

Por otra parte, fue una experiencia muy grata debido a que logré trabajar efectivamente en equipo con mi compañero y el Dr. Gustavo; además, me dio la oportunidad de reflexionar sobre cómo lo que aprendido en mi carrera académica ha sido muy adecuada para llevar a cabo la investigación, ya que al principio dudaba de ello. Por esa razón, me encuentro considerando tomar mis posgrados en áreas relacionadas a la Computación Cuántica; a pesar de que es un área de bajo interés social y poco activa en México.

- Aprendizajes sociales

Daniel Hernández Mota

Como ya se mencionó, no existe un impacto inmediato en la resolución de problemáticas sociales, sin embargo, se puede brindar un acceso directo a la información de calidad, por lo que se realizaron notas acerca del tema, enriqueciendo el conocimiento en general para la divulgación de información. Así mismo, con el proyecto, lo que se busca es encontrar si el sistema físico propuesto puede en efecto generar una computadora cuántica viable, para después pasar a la etapa experimental y posteriormente a lo práctico, para que así pueda existir un cambio directo. Esto, a la larga, puede contribuir a mejorar la economía del país (por la redistribución de recursos utilizados para la ciencia) así como tener un beneficio social por el potencial que tiene una computadora cuántica para resolver problemas que a una computadora normal le tomaría años.

Es imperativo seguir desarrollando la investigación teórica debido a que, aunque no se vean los frutos directamente, tiene el potencial de generar cambios. También el realizar notas de lo mismo sigue dando información que posteriormente se pueda utilizar para mejorar algo. En otras palabras, en un principio, intentar fundamentar la aplicación social de la física teórica puede parecer borroso, no obstante gracias

a la física teórica se tienen los avances tecnológicos que hoy en día son utilizados para la medicina y comunicación.

Karina Chiñas Fuentes

Como ya se mencionó, el tema de Computación Cuántica no es un área del conocimiento muy activa en el país; el que haya personas trabajando en este tema da la oportunidad de que este sea promovido entre más personas y haya cada vez más interesados. Haber trabajado en este tema me dio la oportunidad de ser ese pequeño porcentaje de la población que intenta sacar adelante temas de este tipo de complejidad; principalmente con la redacción de notas en LATEX, cuyo principal objetivo es que ese tipo de conocimiento se vuelva alcanzable para aquellos que no cuentan con altos niveles de razonamiento matemático y conocimiento en física.

Dada la naturaleza del tipo de investigación y las condiciones en la que se encuentra México, los beneficios sociales que se obtienen a partir de este tipo de actividades son muy reducidos y no instantáneos; no obstante, es claro que el beneficio instantáneo se lo lleva el centro de investigación y todos aquellos que colaboran para ella; tanto investigadores como estudiantes, ya que se trabaja en conjunto para proponer ideas y resolver problemáticas. Por otra parte, considero que, si persisto en este ámbito profesional, podré notar los beneficios en grupos sociales que van más allá del centro de investigación. Tal como lo ha logrado el Dr. Gustavo López Velázquez o el Dr. Salvador E. Venegas Andraca.

- Aprendizajes éticos

Daniel Hernández Mota

No hubo una toma de decisiones tan crítica para poder llevar a cabo esta investigación debido a que cometer un error no es tan grave en esta área, entonces siempre se puede detener y regresar al paso anterior, ya que no hay tantos recursos materiales utilizados para la realización de los mismos, haciendo que el impacto económico no sea tan fuerte.

La experiencia de este proyecto de aplicación profesional me lleva a mantener el camino de la investigación para intentar seguir obteniendo resultados a pesar de que parezca una acción difícil de realizar. En concreto ayuda a definir mi profesión, la cual tengo pensada en ejercer ante el beneficio de la vida y la humanidad, intentando mantener la integridad de todos los seres vivos a través del conocimiento teórico buscando de alguna manera una aplicación posible.

Karina Chiñas Fuentes

Considero que las principales actividades que evidenciaron nuestro compromiso con cumplir con nuestras responsabilidades durante el PAP fueron los horarios y espacios; ya que, a mi compañero y a mí, se nos otorgó mucha confianza y libertad para realizar nuestras tareas, tales como el no tener un horario fijo ni fechas límites para entregar avances (principalmente por otras actividades, tanto académicas como personales) y el espacio que se nos otorgó fue en la oficina del Dr. Gustavo. Por lo que, como respuesta, mi compañero y yo siempre tratamos el espacio del Dr. con respeto y entrábamos y salíamos a las horas antes establecidas e intentábamos no tardar más de una semana para tener considerables avances.

Creo que esto muestra, principalmente, nuestro compromiso profesional no sólo con el proyecto, sino con el Doctor y nosotros mismos para salir adelante en la investigación.

- Aprendizajes en lo personal

Daniel Hernández Mota

El PAP me ayudó a identificar que el área de investigación es algo en lo que podría dedicarme sin problema alguno. Me ayudó a determinar que, aunque las cosas parecieran complicadas siempre se puede entenderlas y también se pueden extrapolar conclusiones no triviales con el esfuerzo y la dedicación merecidas. También me hizo entender que es necesario un trabajo en equipo donde se esperan

los resultados de otras personas para poder continuar con el trabajo que necesites, pero no por eso mismo deben actuar por separado, sino que al complementarse pueden surgir resultados más grandes que por individuales no serían fáciles de obtener.

Karina Chiñas Fuentes

Este Proyecto de Aplicación Profesional me ha ayudado, principalmente a darme cuenta de mis capacidades para abordar temas complejos y avanzados como lo es la computación cuántica. Sé que lo que se trabajó en este semestre no contiene todo lo que la computación cuántica ofrece; sin embargo, me ha ayudado a decidir que quiero seguir con mis estudios y me ha dado una idea hacia qué podría orientar mis posgrados e incluso en qué lugares podría laborar.

5. Conclusiones

Se desarrollaron las destrezas necesarias para poder llegar a las simulaciones de computación cuántica en los lenguajes de programación en FORTRAN y PYTHON, obteniendo códigos de alta calidad. Con esto se llegó a la simulación con consideraciones ideales de una computadora cuántica con 1, 2 y 3 qubits, basada en la física de la estructura cristalina del diamante dopado con carbono-13.

No obstante, la redacción de notas para facilitar la accesibilidad a este tipo de información no avanzó como se esperaba debido a que se dejó a un lado por la complejidad y la concentración necesaria para desarrollar los programas, primeramente, en PYTHON utilizando todas las herramientas matemáticas necesarias y secundariamente en FORTRAN 95. El uso de las herramientas tecnológicas como UBUNTU, OVERLEAF y JUPITER fueron aprovechadas para llevar a cabo esta investigación, no obstante, un uso más apropiado de otras plataformas (como GITHUB) podría haber facilitado todavía más el trabajo.

Por ahora el trabajo sigue estando bajo condiciones ideales, por lo que no se ha considerado *ruido* o factores externos que podrían acercar los análisis a un sistema real, por lo que, se tiene pensado implementar *análisis estocásticos* para futuros trabajos.

Por otra parte, solamente se tiene el cómputo realizado para tres qubits, por lo que una generalización de las mismas consideraciones sería la base necesaria para la formación de una *computadora cuántica general*, de “N” qubits; al acoplar la teoría con perturbaciones de tipo aleatorias, se llegaría a tener una buena aproximación de la realidad en la que estaría sometida la computadora cuántica. Creemos que podemos empezar a trabajar en estas consideraciones para la siguiente etapa del PAP y, del mismo modo, poder continuar con la redacción de las notas en LATEX para tener un registro apropiado de toda la información con la que se necesite y genere durante la investigación.

6. Bibliografía

1. **Paredes López, Mireya.** CINVESTAV. *Simulación de Cómputo Cuántico*. [En línea] Departamento de Computación, 10 de Enero de 2007. [Citado el: 06 de Septiembre de 2018.]
<http://www.cs.cinvestav.mx/TesisGraduados/2007/tesisMireyaParedes.pdf>.
2. **Miranda, Nasser Darwish.** nasserdaewish.eu. *Computación Cuántica*. [En línea] Universidad de La Laguna, 2005. [Citado el: 06 de Septiembre de 2018.]
http://nasserdarwish.eu/FILES/Computacion_Cuantica.pdf.
3. **Collazo Rodríguez, Wendolin.** Avance y Perspectiva CINVESTAV. *Resolver problemas de impacto social*. [En línea] 1 de Diciembre de 2017. [Citado el: 06 de Septiembre de 2018.]
<http://ayp.cinvestav.mx/Publicaciones/ArtMID/4126/ArticleID/1493/%E2%80%9CR esolver-problemas-de-impacto-social%E2%80%9D>.
4. **Inc., D-Wave Systems.** DWAVE. *The D-Wave 2000Q™ System*. [En línea] D-Wave Systems Inc. The Quantum Computer Company, 2018. [Citado el: 18 de 10 de 2018.] <https://www.dwavesys.com/d-wave-two-system>.
5. **Tarifa, Enrique Eduardo.** Econ.Unicen. *Teoría de Modelos y Simulación*. [En línea] Universidad Nacional de Jujuy. [Citado el: 06 de Septiembre de 2018.]
https://www.econ.unicen.edu.ar/attachments/1051_TecnicasII Simulacion.pdf.
6. **Gutiérrez Rojas, Ivan R., Peralta Benítez, Hipólito y Fuentes González, Homero C.** Scielo. *Cultura científica y cultura científico investigativa*. [En línea] 2018. [Citado el: 06 de Septiembre de 2018.]
<http://scielo.sld.cu/pdf/hmc/v18n1/hmc03118.pdf>.
7. **Peña, Antonio.** Redalyc. *La investigación científica en México. Estado actual, algunos problemas y perspectivas*. [En línea] enero de 1995. [Citado el: 06 de Septiembre de 2018.] <http://www.redalyc.org/pdf/132/13206702.pdf>. ISSN: 0185-2698.
8. **Scherer, Leopoldo García-Colín.** UNAM. *La ciencia y el Estado*. [En línea] Instituto de Investigaciones Jurídicas de la UNAM. [Citado el: 07 de Septiembre de 2018.] <https://archivos.juridicas.unam.mx/www/bjv/libros/6/2873/11.pdf> .
9. **UNESCO.** Banco Mundial. *Gasto en investigación y desarrollo (% del PIB)*. [En línea] Instituto de Estadística de la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura, 2018. [Citado el: 06 de Septiembre de 2018.]

https://datos.bancomundial.org/indicador/GB.XPD.RSDV.GD.ZS?locations=MX-ZJ&year_high_desc=false.

10. **Olivares Alonso, Emir.** La Jornada. *Recorte presupuestal al Conacyt impactará las becas de posgrado.* [En línea] 22 de Marzo de 2017. [Citado el: 06 de Septiembre de 2018.] <https://www.jornada.com.mx/2017/03/22/sociedad/035n1soc>.

11. **Cassani, Emiliano.** Ciencia UNAM. *Complicado panorama presupuestal para la ciencia en México.* [En línea] Universidad Nacional Autónoma de México, 04 de Febrero de 2018. [Citado el: 06 de Septiembre de 2018.] <http://ciencia.unam.mx/leer/705/complicado-panorama-presupuestal-para-la-ciencia-en-mexico>.

12. **Maya Higuera, Victor.** Sociedad Mexicana de Física. *Convocatoria Veranos Científicos en Óptica Cuántica, Materia Fría e Información Cuántica 2018.* [En línea] Sociedad Mexicana de Física, 21 de Mayo de 2018. [Citado el: 06 de Septiembre de 2018.] <https://www.smf.mx/convocatoria-veranos-cientificos-en-optica-cuantica-materia-fria-e-informacion-cuantica-2018/>.

13. **Commission, United States Securities and Exchange.** FORM D. *Notice of Exempt Offering of Securities.* Washington D.C. : s.n., 2018. OMB APPROVAL: 3235-0076.

14. **Venegas Andraca, Salvador E.** *Workshop on Quantum Computation using IBM Q.* Ciudad de México : Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, 2018. 1.

15. **Zwiebach, Barton.** EDX. *Overview of Quantum Mechanics: Necessity of Complex Numbers.* [En línea] Massachusetts Institute of Technology, 2017. [Citado el: 6 de Octubre de 2018.] https://courses.edx.org/courses/course-v1:MITx+8.04.1x+3T2017/courseware/week1/lecture1/6?activate_block_id=block-v1%3AMITx%2B8.04.1x%2B3T2017%2Btype%40vertical%2Bblock%40v-vid1-3}.

16. **Gasiorowicz, Stephen.** Quantum Mechanics. *Quantum Mechanics.* United States of America : Wiley, 2003.

17. **Griffiths, David J.** Quantum Mechanics. *Quantum Mechanics.* United Kingdom : Cambridge University Press, 2017.

18. **Vazirani, Umesh.** Qubits, Quantum Mechanics, and Computers. California : University of California, Berkeley, 2012.

19. **Michael A. Nielsen, Isaac I. Chuang.** *Quantum Computation and Quantum Information.* United Kingdom : Cambridge University Press, 2010. 978-1-197-00217-3.

20. **López, Gustavo.** Notas de Computación Cuántica. Guadalajara, Jalisco : s.n., 2018. Curso 3.
21. **Chiñas, K. Hernández, D.** Autoridad Propia. Guadalajara : s.n., 2018.
22. **García, Angel Franco.** Sistema de Ecuaciones Diferenciales de Primer Orden. *Procedimientos Numéricos en Lenguaje Java*. [En línea] Universidad del País Vasco, 2000. [Citado el: 18 de Octubre de 2018.]
<http://www.sc.ehu.es/sbweb/fisica/cursoJava/numerico/eDiferenciales/rungeKutta/rungeKutta1.htm>.
23. **Fausett, Laurene V.** *Numerical Methods Algorithms and Applications*. New Jersey : Prentice Hall, 2003. 0-13-031400-5.
24. **López, G. Lara, L.** *Numerical Simulation of a Controlled-Controlled-Not (CCN) quantum gate in a chain of three interacting nuclear spins systems*. México : Journal of Physics B: Atomic, Molecular and Optical Physics, 2006, Vol. 39. doi:10.1088/0953-4075/39/18/019.
25. **Carlson, Stephan C.** Britannica Academic. *Hilber Space*. [En línea] Encyclopædia Britannica, 11 de Agosto de 2018. [Citado el: 7 de Octubre de 2018.]
<https://academic-eb-com.ezproxy.iteso.mx/levels/collegiate/article/Hilbert-space/384376>.

Anexos

Anexo 1. Código en PYTHON para un qubit

```
"""
PROGRAMA QUE RESUELVE NUMERICAMENTE LAS FUNCIONES  $D_0(t)$  y  $D_1(t)$ 
QUE SE OBTIENEN AL TRATAR LA ECUACION DEL ESQUEMA DE INTERACCIÓN
PARA EL CARBONO C13 EN DIAMANTE BAJO LA INFLUENCIA DE UN CAMPO
MAGNETICO

27/09/2018 - 15/11/2018
1PQubit.py
"""

import numpy as np
from numpy import sin, cos, sqrt
import matplotlib.pyplot as plt
from matplotlib.pyplot import savefig
import matplotlib.path_effects as path_effects

# Determinar el número de pasos de acuerdo a las condiciones.
ti = 0
tf = 5
h = 0.01
n = int((tf-ti)/h)

# Constantes del sistema
w0 = 0.5

U = 1/w0
w = 100/w0
wt = 100/w0

phi = 0

E0 = -wt*0.5
E1 = wt*0.5

# Arreglos para solución numerica
vX0 = np.zeros(n)
vY0 = np.zeros(n)
vX1 = np.zeros(n)
vY1 = np.zeros(n)

cX0 = np.zeros(n)
cY0 = np.zeros(n)
cX1 = np.zeros(n)
cY1 = np.zeros(n)

T = np.zeros(n)
T[0] = ti

# Condiciones iniciales
vX0[0] = sqrt(9./200.)
vY0[0] = (sqrt(2.))/5.

FO-DGA-CPAP-0017
```

```

vX1[0] = (sqrt(6.))/4.
vY1[0] = (sqrt(2.))/2.

cX0[0] = vX0[0]*cos(E0*T[0]) - vY0[0]*sin(E0*T[0])
cY0[0] = vX0[0]*sin(E0*T[0]) + vY0[0]*cos(E0*T[0])

cX1[0] = vX1[0]*cos(E1*T[0]) - vY1[0]*sin(E1*T[0])
cY1[0] = vX1[0]*sin(E1*T[0]) + vY1[0]*cos(E1*T[0])

# Funciones que describen el sistema
alpha = lambda t: ((wt - w)*t) - phi

dX0 = lambda t, X1, Y1: (U*0.5)*( np.sin(alpha(t))*X1 -
np.cos(alpha(t))*Y1 )
dY0 = lambda t, X1, Y1: (U*0.5)*( np.cos(alpha(t))*X1 +
np.sin(alpha(t))*Y1 )
dX1 = lambda t, X0, Y0: -(U*0.5)*( np.sin(alpha(t))*X0 +
np.cos(alpha(t))*Y0 )
dY1 = lambda t, X0, Y0: (U*0.5)*( np.cos(alpha(t))*X0 -
np.sin(alpha(t))*Y0 )

D0 = np.zeros(n)
D1 = np.zeros(n)

N = np.zeros(n)

# Constantes de Runge-Kutta
# Nota: K hace referencia a reales y L a Complejos
for i in range(0,n-1):

    T[i+1] = T[i] + h

    Koa = h*dX0(T[i], vX1[i], vY1[i])
    Loa = h*dY0(T[i], vX1[i], vY1[i])
    Kia = h*dX1(T[i], vX0[i], vY0[i])
    Lia = h*dY1(T[i], vX0[i], vY0[i])

    Kob = h*dX0(T[i] + h/2, vX1[i] + Kia/2, vY1[i] + Lia/2)
    Lob = h*dY0(T[i] + h/2, vX1[i] + Kia/2, vY1[i] + Lia/2)
    Kib = h*dX1(T[i] + h/2, vX0[i] + Koa/2, vY0[i] + Loa/2)
    Lib = h*dY1(T[i] + h/2, vX0[i] + Koa/2, vY0[i] + Loa/2)

    Koc = h*dX0(T[i] + h/2, vX1[i] + Kib/2, vY1[i] + Lib/2)
    Loc = h*dY0(T[i] + h/2, vX1[i] + Kib/2, vY1[i] + Lib/2)
    Kic = h*dX1(T[i] + h/2, vX0[i] + Kob/2, vY0[i] + Lob/2)
    Lic = h*dY1(T[i] + h/2, vX0[i] + Kob/2, vY0[i] + Lob/2)

    Kod = h*dX0(T[i] + h, vX1[i] + Kic, vY1[i] + Lic)
    Lod = h*dY0(T[i] + h, vX1[i] + Kic, vY1[i] + Lic)
    Kid = h*dX1(T[i] + h, vX0[i] + Koc, vY0[i] + Loc)
    Lid = h*dY1(T[i] + h, vX0[i] + Koc, vY0[i] + Loc)

# Soluciones por Runge-Kutta
vX0[i+1] = vX0[i] + (1/6)*(Koa + 2*Kob + 2*Koc + Kod)
vY0[i+1] = vY0[i] + (1/6)*(Loa + 2*Lob + 2*Loc + Lod)

```

```

vX1[i+1] = vX1[i] + (1/6)*(Kia + 2*Kib + 2*Kic + Kid)
vY1[i+1] = vY1[i] + (1/6)*(Lia + 2*Lib + 2*Lic + Lid)

cX0[i+1] = vX0[i+1]*cos(E0*T[i+1]) - vY0[i+1]*sin(E0*T[i+1])
cY0[i+1] = vX0[i+1]*sin(E0*T[i+1]) + vY0[i+1]*cos(E0*T[i+1])

cX1[i+1] = vX1[i+1]*cos(E1*T[i+1]) - vY1[i+1]*sin(E1*T[i+1])
cY1[i+1] = vX1[i+1]*sin(E1*T[i+1]) + vY1[i+1]*cos(E1*T[i+1])

D0[i] = (vX0[i])**2 + (vY0[i])**2
D1[i] = (vX1[i])**2 + (vY1[i])**2
N[i] = D0[i] + D1[i]

# Se completan los vectores
D0[-1] = (vX0[-1])**2 + (vY0[-1])**2
D1[-1] = (vX1[-1])**2 + (vY1[-1])**2
N[-1] = D1[-1] + D0[-1]

# Graficar la comprobacion del sistema.
sstyle = "seaborn-poster"
plt.style.use(sstyle)
plt.rc('text', usetex=True)
plt.rc('font', family='serif')
plt.figure(figsize = (20,10))

wt = 100
w = 100
U = 1
phi = 0

path = "/PyFigures/1QubitFig/"
file_name = path + "X0_" + "%.2f"%vX0[0] + " Y0_" + "%.2f"%vY0[0] + \
            " X1_" + "%.2f"%vX1[0] + " Y1_" + "%.2f"%vY1[0] + \
            " wt_" + "%.2f"%wt + " w_" + "%.2f"%w + \
            " U_" + "%.2f"%U + " phi_" + "%.2f"%phi + \
            " h_" + "%.4f"%h

plt.plot(T, N, label = "Total Probability", linewidth = 5.0)
plt.plot(T, D0, label = "$|C_{0}(t)|^2$")
plt.plot(T, D1, label = "$|C_{1}(t)|^2$")

plt.title("One-Qubit Probabilities")
plt.xlabel("Time (s)")
plt.ylabel("Probability")

plt.legend(loc = 0)
plt.grid()

#plt.savefig( file_name + ".png", format = "png")
plt.show()

XMAX = np.array( [ max(cX0), max(cX1) ] )
YMAX = np.array( [ max(cY0), max(cY1) ] )
TMAX = max( np.concatenate( (XMAX,YMAX), axis = None ) ) + 0.1

```

```

XMIN = np.array( [ min(cX0), min(cX1) ] )
YMIN = np.array( [ min(cY0), min(cY1) ] )
TMIN = min( np.concatenate( (XMIN,YMIN), axis = None ) ) - 0.1

COMP = input( "Do you want to see the components of the wave function?
[Y/N]: ")

if(COMP == 'N' or COMP == 'n' or COMP == 'No' or COMP == 'NO' or COMP ==
'no' or COMP == 'Nope' or COMP == 'nope'):
    pass

elif (COMP == 'Y' or COMP == 'y' or COMP == 'Yes' or COMP == 'YES' or
COMP == 'yes'):

    # GRAFICA DE LOS COEFICIENTES DE LA FUNCION DE ONDA DEL SISTEMA
    # -----
    # PRIMER COEFICIENTE
    # -----

    plt.figure(figsize = (20,10))
    plt.ylim(top = TMAX)
    plt.ylim(bottom = TMIN)

    plt.plot(T, cX0, label = "$X_{0}$ - Real ")
    plt.plot(T, cY0,label = "$Y_{0}$ - Imaginary")
    plt.title( "Components of the Coefficient $C_{0}(t)$")
    plt.xlabel("Time (s)")
    plt.ylabel(" ")
    plt.legend(loc=0)
    plt.grid()
    plt.show()

    # -----
    # SEGUNDO COEFICIENTE
    # -----

    plt.figure(figsize = (20,10))
    plt.ylim(top = TMAX)
    plt.ylim(bottom = TMIN)

    plt.plot(T, cX1, label = "$X_{1}$ - Real ")
    plt.plot(T, cY1, label = "$Y_{1}$ - Imaginary")
    plt.title( "Components of the Coefficient $C_{1}(t)$")
    plt.xlabel("Time (s)")
    plt.ylabel(" ")
    plt.legend(loc=0)
    plt.grid()
    plt.show()

```

Anexo 2. Código en PYTHON para dos qubits

```
"""
PROGRAMA QUE RESUELVE NUMERICAMENTE LAS FUNCIONES  $D_{00}(t)$ ,  $D_{01}(t)$ ,
 $D_{10}(t)$  y  $D_{11}(t)$  QUE SE OBTIENEN AL TRATAR LA ECUACION DEL ESQUEMA DE
INTERACCION PARA DOS QUBITS DE CARBONO C13 EN UN ENLACE DE DIAMANTE BAJO LA
INFLUENCIA DE UN CAMPO MAGNETICO.

04/10/2018 - 15/11/2018
2PQubit.py
"""

import numpy as np
from numpy import sin, cos, sqrt
import matplotlib.pyplot as plt
from matplotlib.pyplot import savefig
import matplotlib.path_effects as path_effects

#Constantes del sistema
w0 = 1/np.pi

U = 1/w0
w = 100.0/w0
w1 = 100.0/w0
w2 = 100.0/w0
J = 0.001/w0

phi = 0.000

# Determinar el numero de pasos de acuerdo a las condiciones.
ti = 0
tf = 10
h = 0.001
n = int((tf-ti)/h)

# Energias del sistema por unidad de h-barra

E = lambda a2,a1: (-1/2)* ( 0.5*J)*((-1)**(a1+a2)) + w1*(-1)**a1 + w2*(-1)**a2 )

E00 = E(0,0)
E01 = E(0,1)
E10 = E(1,0)
E11 = E(1,1)

# Arreglos para solucion numerica
vX00, vY00 = np.zeros(n), np.zeros(n)
vX01, vY01 = np.zeros(n), np.zeros(n)
vX10, vY10 = np.zeros(n), np.zeros(n)
vX11, vY11 = np.zeros(n), np.zeros(n)

cX00, cY00 = np.zeros(n), np.zeros(n)
cX01, cY01 = np.zeros(n), np.zeros(n)
cX10, cY10 = np.zeros(n), np.zeros(n)
cX11, cY11 = np.zeros(n), np.zeros(n)
```

```

D00 = np.zeros(n)
D01 = np.zeros(n)
D10 = np.zeros(n)
D11 = np.zeros(n)

N      = np.zeros(n)
T      = np.zeros(n)
T[0] = ti

# Condiciones inidicales del esquema de interaccion
vX00[0] = sqrt(4/4)
vY00[0] = sqrt(0/4)

vX01[0] = sqrt(0/4)
vY01[0] = sqrt(0/4)

vX10[0] = sqrt(0/4)
vY10[0] = sqrt(0/4)

vX11[0] = sqrt(0/4)
vY11[0] = sqrt(0/4)

# Condiciones inidicales de la funcion de onda
cX00[0] = vX00[0]*cos(E00*T[0]) - vY00[0]*sin(E00*T[0])
cY00[0] = vX00[0]*sin(E00*T[0]) + vY00[0]*cos(E00*T[0])

cX01[0] = vX01[0]*cos(E01*T[0]) - vY01[0]*sin(E01*T[0])
cY01[0] = vX01[0]*sin(E01*T[0]) + vY01[0]*cos(E01*T[0])

cX10[0] = vX10[0]*cos(E10*T[0]) - vY10[0]*sin(E10*T[0])
cY10[0] = vX10[0]*sin(E10*T[0]) + vY10[0]*cos(E10*T[0])

cX11[0] = vX11[0]*cos(E11*T[0]) - vY11[0]*sin(E11*T[0])
cY11[0] = vX11[0]*sin(E11*T[0]) + vY11[0]*cos(E11*T[0])

# angulos que describen el sistema
theta = lambda t: w*t + phi
ooA01 = lambda t: -(w1+(J/2))*t + theta(t)
ooA10 = lambda t: -(w2+(J/2))*t + theta(t)
iiA01 = lambda t: (w2-(J/2))*t - theta(t)
iiA10 = lambda t: (w1-(J/2))*t - theta(t)

# Ecuaciones diferenciales que describen el sistema.
dX00 = lambda t, X01, X10, Y01, Y10: (U/2)*(- X01*np.sin(ooA01(t)) -
Y01*np.cos(ooA01(t)) - X10*np.sin(ooA10(t)) - Y10*np.cos(ooA10(t)) )
dY00 = lambda t, X01, X10, Y01, Y10: (U/2)*( X01*np.cos(ooA01(t)) -
Y01*np.sin(ooA01(t)) + X10*np.cos(ooA10(t)) - Y10*np.sin(ooA10(t)) )

dX11 = lambda t, X01, X10, Y01, Y10: (U/2)*(- X01*np.sin(iiA01(t)) -
Y01*np.cos(iiA01(t)) - X10*np.sin(iiA10(t)) - Y10*np.cos(iiA10(t)) )
dY11 = lambda t, X01, X10, Y01, Y10: (U/2)*( X01*np.cos(iiA01(t)) -
Y01*np.sin(iiA01(t)) + X10*np.cos(iiA10(t)) - Y10*np.sin(iiA10(t)) )

dX01 = lambda t, X00, X11, Y00, Y11: (U/2)*( X00*np.sin(ooA01(t)) -
Y00*np.cos(ooA01(t)) + X11*np.sin(iiA01(t)) - Y11*np.cos(iiA01(t)) )

```

```
dY01 = lambda t, X00, X11, Y00, Y11: (U/2)*( X00*np.cos(ooA01(t)) +
Y00*np.sin(ooA01(t)) + X11*np.cos(iiA01(t)) + Y11*np.sin(iiA01(t)) )
```

```
dX10 = lambda t, X00, X11, Y00, Y11: (U/2)*( X00*np.sin(ooA10(t)) -
Y00*np.cos(ooA10(t)) + X11*np.sin(iiA10(t)) - Y11*np.cos(iiA10(t)) )
dY10 = lambda t, X00, X11, Y00, Y11: (U/2)*( X00*np.cos(ooA10(t)) +
Y00*np.sin(ooA10(t)) + X11*np.cos(iiA10(t)) + Y11*np.sin(iiA10(t)) )
```

```
# Constantes de Runge-Kutta
# Nota: K hace referencia a reales y L a Complejos
for i in range(0,n-1):
```

```
    T[i+1] = T[i] + h
```

```
    # Constantes (referenciadas por la letra a):
```

```
    K00a = h*dX00(T[i], vX01[i], vX10[i], vY01[i], vY10[i])
    L00a = h*dY00(T[i], vX01[i], vX10[i], vY01[i], vY10[i])
    Kiia = h*dX11(T[i], vX01[i], vX10[i], vY01[i], vY10[i])
    Liia = h*dY11(T[i], vX01[i], vX10[i], vY01[i], vY10[i])
```

```
    K01a = h*dX01(T[i], vX00[i], vX11[i], vY00[i], vY11[i])
    L01a = h*dY01(T[i], vX00[i], vX11[i], vY00[i], vY11[i])
    K10a = h*dX10(T[i], vX00[i], vX11[i], vY00[i], vY11[i])
    L10a = h*dY10(T[i], vX00[i], vX11[i], vY00[i], vY11[i])
```

```
    # Constantes (referenciadas por la letra b):
```

```
    K00b = h*dX00(T[i] + h/2, vX01[i] + K01a/2, vX10[i] + K10a/2,
vY01[i] + L01a/2, vY10[i] + L10a/2)
    L00b = h*dY00(T[i] + h/2, vX01[i] + K01a/2, vX10[i] + K10a/2,
vY01[i] + L01a/2, vY10[i] + L10a/2)
    K11b = h*dX11(T[i] + h/2, vX01[i] + K01a/2, vX10[i] + K10a/2,
vY01[i] + L01a/2, vY10[i] + L10a/2)
    L11b = h*dY11(T[i] + h/2, vX01[i] + K01a/2, vX10[i] + K10a/2,
vY01[i] + L01a/2, vY10[i] + L10a/2)
```

```
    K01b = h*dX01(T[i] + h/2, vX00[i] + K00a/2, vX11[i] + Kiia/2,
vY00[i] + L00a/2, vY11[i] + Liia/2)
    L01b = h*dY01(T[i] + h/2, vX00[i] + K00a/2, vX11[i] + Kiia/2,
vY00[i] + L00a/2, vY11[i] + Liia/2)
    K10b = h*dX10(T[i] + h/2, vX00[i] + K00a/2, vX11[i] + Kiia/2,
vY00[i] + L00a/2, vY11[i] + Liia/2)
    L10b = h*dY10(T[i] + h/2, vX00[i] + K00a/2, vX11[i] + Kiia/2,
vY00[i] + L00a/2, vY11[i] + Liia/2)
```

```
    # Constantes (referenciadas por la letra c):
```

```
    K00c = h*dX00(T[i] + h/2, vX01[i] + K01b/2, vX10[i] + K10b/2,
vY01[i] + L01b/2, vY10[i] + L10b/2)
    L00c = h*dY00(T[i] + h/2, vX01[i] + K01b/2, vX10[i] + K10b/2,
vY01[i] + L01b/2, vY10[i] + L10b/2)
    K11c = h*dX11(T[i] + h/2, vX01[i] + K01b/2, vX10[i] + K10b/2,
vY01[i] + L01b/2, vY10[i] + L10b/2)
    L11c = h*dY11(T[i] + h/2, vX01[i] + K01b/2, vX10[i] + K10b/2,
vY01[i] + L01b/2, vY10[i] + L10b/2)
```

```
    K01c = h*dX01(T[i] + h/2, vX00[i] + K00b/2, vX11[i] + K11b/2,
vY00[i] + L00b/2, vY11[i] + L11b/2)
```

```

    L01c = h*dY01(T[i] + h/2, vX00[i] + K00b/2, vX11[i] + K11b/2,
vY00[i] + L00b/2, vY11[i] + L11b/2)
    K10c = h*dX10(T[i] + h/2, vX00[i] + K00b/2, vX11[i] + K11b/2,
vY00[i] + L00b/2, vY11[i] + L11b/2)
    L10c = h*dY10(T[i] + h/2, vX00[i] + K00b/2, vX11[i] + K11b/2,
vY00[i] + L00b/2, vY11[i] + L11b/2)

# Constantes (referenciadas por la letra d):
    K00d = h*dX00(T[i] + h, vX01[i] + K01c, vX10[i] + K10c, vY01[i] +
L01c, vY10[i] + L10c)
    L00d = h*dY00(T[i] + h, vX01[i] + K01c, vX10[i] + K10c, vY01[i] +
L01c, vY10[i] + L10c)
    K11d = h*dX11(T[i] + h, vX01[i] + K01c, vX10[i] + K10c, vY01[i] +
L01c, vY10[i] + L10c)
    L11d = h*dY11(T[i] + h, vX01[i] + K01c, vX10[i] + K10c, vY01[i] +
L01c, vY10[i] + L10c)

    K01d = h*dX01(T[i] + h, vX00[i] + K00c, vX11[i] + K11c, vY00[i] +
L00c, vY11[i] + L11c)
    L01d = h*dY01(T[i] + h, vX00[i] + K00c, vX11[i] + K11c, vY00[i] +
L00c, vY11[i] + L11c)
    K10d = h*dX10(T[i] + h, vX00[i] + K00c, vX11[i] + K11c, vY00[i] +
L00c, vY11[i] + L11c)
    L10d = h*dY10(T[i] + h, vX00[i] + K00c, vX11[i] + K11c, vY00[i] +
L00c, vY11[i] + L11c)

# Soluciones para el esquema de interaccion por Runge-Kutta
    vX00[i+1] = vX00[i] + (1/6)*(K00a + 2*K00b + 2*K00c + K00d)
    vY00[i+1] = vY00[i] + (1/6)*(L00a + 2*L00b + 2*L00c + L00d)

    vX01[i+1] = vX01[i] + (1/6)*(K01a + 2*K01b + 2*K01c + K01d)
    vY01[i+1] = vY01[i] + (1/6)*(L01a + 2*L01b + 2*L01c + L01d)

    vX10[i+1] = vX10[i] + (1/6)*(K10a + 2*K10b + 2*K10c + K10d)
    vY10[i+1] = vY10[i] + (1/6)*(L10a + 2*L10b + 2*L10c + L10d)

    vX11[i+1] = vX11[i] + (1/6)*(K11a + 2*K11b + 2*K11c + K11d)
    vY11[i+1] = vY11[i] + (1/6)*(L11a + 2*L11b + 2*L11c + L11d)

# Soluciones para la funcion de onda
    cX00[i+1] = vX00[i+1]*cos(E00*T[i+1]) - vY00[i+1]*sin(E00*T[i+1])
    cY00[i+1] = vX00[i+1]*sin(E00*T[i+1]) + vY00[i+1]*cos(E00*T[i+1])

    cX01[i+1] = vX01[i+1]*cos(E01*T[i+1]) - vY01[i+1]*sin(E01*T[i+1])
    cY01[i+1] = vX01[i+1]*sin(E01*T[i+1]) + vY01[i+1]*cos(E01*T[i+1])

    cX10[i+1] = vX10[i+1]*cos(E10*T[i+1]) - vY10[i+1]*sin(E10*T[i+1])
    cY10[i+1] = vX10[i+1]*sin(E10*T[i+1]) + vY10[i+1]*cos(E10*T[i+1])

    cX11[i+1] = vX11[i+1]*cos(E11*T[i+1]) - vY11[i+1]*sin(E11*T[i+1])
    cY11[i+1] = vX11[i+1]*sin(E11*T[i+1]) + vY11[i+1]*cos(E11*T[i+1])

# Determinacion de probabilidades

    D00[i] = (cX00[i])**2 + (cY00[i])**2
    D01[i] = (cX01[i])**2 + (cY01[i])**2

```

```

D10[i] = (cX10[i])**2 + (cY10[i])**2
D11[i] = (cX11[i])**2 + (cY11[i])**2

# Comprobacion de normalidad
N[i] = D00[i] + D01[i] + D10[i] + D11[i]

D00[-1] = (cX00[-1])**2 + (cY00[-1])**2
D01[-1] = (cX01[-1])**2 + (cY01[-1])**2
D10[-1] = (cX10[-1])**2 + (cY10[-1])**2
D11[-1] = (cX11[-1])**2 + (cY11[-1])**2

N[-1] = D00[-1] + D01[-1] + D10[-1] + D11[-1]

# Graficar la comprobacion del sistema.

sstyle = "seaborn-poster"
plt.style.use(sstyle)
plt.rc('text', usetex=True)
plt.rc('font', family='serif')
plt.figure(figsize = (20,10))

file_name = "X00_" + "%.2f"%vX00[0] + " Y00_" + "%.2f"%vY00[0] + \
" X01_" + "%.2f"%vX01[0] + " Y01_" + "%.2f"%vY01[0] + \
" X10_" + "%.2f"%vX10[0] + " Y10_" + "%.2f"%vY10[0] + \
" X11_" + "%.2f"%vX11[0] + " Y11_" + "%.2f"%vY11[0] + \
" U_" + "%.2f"%U + " w_" + "%.2f"%w + \
" w1_" + "%.2f"%w1 + " w2_" + "%.2f"%w2 + \
" J_" + "%.2f"%J + " phi_" + "%.2f"%phi + \
" h_" + "%.4f"%h

plt.plot(T, N, label = "Total Probability")
plt.plot(T, D00, label = "$|C_{00}(t)|^2$")
plt.plot(T, D01, label = "$|C_{01}(t)|^2$")
plt.plot(T, D10, label = "$|C_{10}(t)|^2$")
plt.plot(T, D11, label = "$|C_{11}(t)|^2$")

plt.title("Two-Qubit Probabilities")
plt.xlabel("Time (s)")
plt.ylabel("Probability")

plt.legend(loc = 0)
plt.grid()

#plt.savefig( file_name + ".png", format = "png")
plt.show()

# COEFICIENTES DE LA FUNCION DE ONDA

XMAX = np.array( [ max(cX00), max(cX01), max(cX10), max(cX11) ] )
YMAX = np.array( [ max(cY00), max(cY01), max(cY10), max(cY11) ] )
TMAX = max( np.concatenate( (XMAX,YMAX), axis = None ) ) + 0.1

XMIN = np.array( [ min(cX00), min(cX01), min(cX10), min(cX11) ] )

```

```

YMIN = np.array( [ min(cY00), min(cY01), min(cY10), min(cY11) ] )
TMIN = min( np.concatenate( (XMIN,YMIN), axis = None ) ) - 0.1

COMP = input( "Do you want to see the components of the wave function?
[Y/N]: ")

if(COMP == 'N' or COMP == 'n' or COMP == 'No' or COMP == 'NO' or COMP ==
'no' or COMP == 'Nope' or COMP == 'nope'):
    pass

elif (COMP == 'Y' or COMP == 'y' or COMP == 'Yes' or COMP == 'YES' or
COMP == 'yes'):

    # -----
    # PRIMER COEFICIENTE
    # -----

    plt.figure(figsize = (20,10))
    plt.ylim(top = TMAX)
    plt.ylim(bottom = TMIN)

    plt.plot(T, cX00, label = "$X_{00}$ - Real ")
    plt.plot(T, cY00, label = "$Y_{00}$ - Imaginary")
    plt.title( "Components of the Coeficiente $C_{00}(t)$")
    plt.xlabel("Time (s)")
    plt.ylabel("Probability")
    plt.legend(loc=0)
    plt.grid()
    plt.show()

    # -----
    # SEGUNDO COEFICIENTE
    # -----

    plt.figure(figsize = (20,10))
    plt.ylim(top = TMAX)
    plt.ylim(bottom = TMIN)

    plt.plot(T, cX01, label = "$X_{01}$ - Real ")
    plt.plot(T, cY01, label = "$Y_{01}$ - Imaginary")
    plt.title( "Components of the Coeficiente $C_{01}(t)$")
    plt.xlabel("Time (s)")
    plt.ylabel("Probability")
    plt.legend(loc=0)
    plt.grid()
    plt.show()

    # -----
    # TERCER COEFICIENTE
    # -----

    plt.figure(figsize = (20,10))
    plt.ylim(top = TMAX)
    plt.ylim(bottom = TMIN)

    plt.plot(T, cX10, label = "$X_{10}$ - Real ")
    plt.plot(T, cY10, label = "$Y_{10}$ - Imaginary")

```

```

plt.title( "Components of the Coeficient $C_{10}(t)$")
plt.xlabel("Time (s)")
plt.ylabel("Probability")
plt.legend(loc=0)
plt.grid()
plt.show()
# -----
# CUARTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX11, label = "$X_{11}$ - Real ")
plt.plot(T, cY11, label = "$Y_{11}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{11}(t)$")
plt.xlabel("Time (s)")
plt.ylabel("Probability")
plt.legend(loc=0)
plt.grid()
plt.show()

```

Anexo 3. Código en PYTHON para tres qubits

```
"""
PROGRAMA QUE RESUELVE NUMERICAMENTE LAS FUNCIONES D_000(t), ..., D_111(t)
QUE SE OBTIENEN AL TRATAR LA ECUACION DEL ESQUEMA DE INTERACCION PARA TRES
QUBITS DE CARBONO C13 EN UN ENLACE DE DIAMANTE BAJO LA INFLUENCIA DE UN
CAMPO MAGNETICO.

23/10/2018 - 16/11/2018
3PQubit.py
"""

import numpy as np
from numpy import sin, cos, sqrt, pi
import matplotlib.pyplot as plt
from matplotlib.pyplot import savefig
import matplotlib.path_effects as path_effects

# CONSTANTES DEL SISTEMA

w0 = 1/pi # FRECUENCIA DEL TIEMPO TAO

U = 0.1/w0 # FRECUENCIA DE RABI
w1 = 100/w0 # FRECUENCIA DE LARMOR DEL PRIMER QUBIT
w2 = 150/w0 # FRECUENCIA DE LARMOR DEL SEGUNDO QUBIT
w3 = 200/w0 # FRECUENCIA DE LARMOR DEL TERCER QUBIT
J = 5.0/w0 # FACTOR DE INTERACCION ENTRE PRIMEROS VECINOS (FRECUENCIA)
Jp = 0.8/w0 # FACTOR DE INTERACCION ENTRE SEGUNDOS VECINOS (FRECUENCIA)

phi = 0.0 # FASE DEL CAMPO MAGNETICO

# DELIMITACIONES DEL METODO NUMERICOC.
ti = 0.0 # TIEMPO INICIAL: siempre cero
tf = 50.0 # TIEMPO FINAL
h = 0.01 # TAMANHO DE PASO
n = int((tf-ti)/h) # NUMERO DE ITERACIONES

# ENERGIA DEL SISTEMA POR CADA UNIDAD h-BARRA.
E_hbar = lambda a3, a2, a1: (-1/2)*( w1*((-1)**a1) + w2*((-1)**a2) +
w3*((-1)**a3) + 0.5*J*( ( (-1)**(a1+a2) ) + ( (-1)**(a2+a3) ) ) +
0.5*Jp*( (-1)**(a1+a3) ) )

E = np.zeros(8)
E[0] = E_hbar(0,0,0) #ENERGIA DEL ESTADO 000
E[1] = E_hbar(0,0,1) #ENERGIA DEL ESTADO 001
E[2] = E_hbar(0,1,0) #ENERGIA DEL ESTADO 010
E[3] = E_hbar(0,1,1) #ENERGIA DEL ESTADO 011
E[4] = E_hbar(1,0,0) #ENERGIA DEL ESTADO 100
E[5] = E_hbar(1,0,1) #ENERGIA DEL ESTADO 101
E[6] = E_hbar(1,1,0) #ENERGIA DEL ESTADO 110
E[7] = E_hbar(1,1,1) #ENERGIA DEL ESTADO 111

Wij = lambda i, j: ( E[i-1] - E[j-1] )/w0
w = Wij(5,1) # FRECUENCIA DEL CAMPO MAGNETICO
FO-DGA-CPAP-0017
```

```

theta = lambda t: w*t + phi # Angulo del campo magnetico
B      = lambda t, i, j: Wij(i,j)*t + theta(t)

# ARREGLOS PARA SOLUCION NUMERICA

vX1, vY1 = np.zeros(n), np.zeros(n)
vX2, vY2 = np.zeros(n), np.zeros(n)
vX3, vY3 = np.zeros(n), np.zeros(n)
vX4, vY4 = np.zeros(n), np.zeros(n)
vX5, vY5 = np.zeros(n), np.zeros(n)
vX6, vY6 = np.zeros(n), np.zeros(n)
vX7, vY7 = np.zeros(n), np.zeros(n)
vX8, vY8 = np.zeros(n), np.zeros(n)

cX1, cY1 = np.zeros(n), np.zeros(n)
cX2, cY2 = np.zeros(n), np.zeros(n)
cX3, cY3 = np.zeros(n), np.zeros(n)
cX4, cY4 = np.zeros(n), np.zeros(n)
cX5, cY5 = np.zeros(n), np.zeros(n)
cX6, cY6 = np.zeros(n), np.zeros(n)
cX7, cY7 = np.zeros(n), np.zeros(n)
cX8, cY8 = np.zeros(n), np.zeros(n)

D1 = np.zeros(n)
D2 = np.zeros(n)
D3 = np.zeros(n)
D4 = np.zeros(n)
D5 = np.zeros(n)
D6 = np.zeros(n)
D7 = np.zeros(n)
D8 = np.zeros(n)

N    = np.zeros(n)
T    = np.zeros(n)

# CONDICIONES INICIALES DEL SISTEMA DE ECUACIONES DEL MARCO DE
INTERACCION
# Nota: tambien cumplen con las condiciones de la funcion de onda

vX1[0], vY1[0] = 1, 0
vX2[0], vY2[0] = 0, 0
vX3[0], vY3[0] = 0, 0
vX4[0], vY4[0] = 0, 0
vX5[0], vY5[0] = 0, 0
vX6[0], vY6[0] = 0, 0
vX7[0], vY7[0] = 0, 0
vX8[0], vY8[0] = 0, 0

# SISTEMA DE ECUACIONES DIFERENCIALES DEL MARCO DE INTERACCION

dX1 = lambda t, X2, Y2, X3, Y3, X5, Y5: (U/2)*(-X2*sin(B(t,1,2)) + -
Y2*cos(B(t,1,2)) + -X3*sin(B(t,1,3)) + -Y3*cos(B(t,1,3)) + -
X5*sin(B(t,1,5)) + -Y5*cos(B(t,1,5)) )
dY1 = lambda t, X2, Y2, X3, Y3, X5, Y5: (U/2)*( X2*cos(B(t,1,2)) + -
Y2*sin(B(t,1,2)) + X3*cos(B(t,1,3)) + -Y3*sin(B(t,1,3))
+ X5*cos(B(t,1,5)) + -Y5*sin(B(t,1,5)) )

```

```

dX2 = lambda t, X1, Y1, X4, Y4, X6, Y6: (U/2)*( X1*sin(B(t,1,2)) + -
Y1*cos(B(t,1,2)) + -X4*sin(B(t,2,4)) + -Y4*cos(B(t,2,4)) + -
X6*sin(B(t,2,6)) + -Y6*cos(B(t,2,6)) )
dY2 = lambda t, X1, Y1, X4, Y4, X6, Y6: (U/2)*( X1*cos(B(t,1,2)) +
Y1*sin(B(t,1,2)) + X4*cos(B(t,2,4)) + -Y4*sin(B(t,2,4))
+ X6*cos(B(t,2,6)) + -Y6*sin(B(t,2,6)) )

```

```

dX3 = lambda t, X1, Y1, X4, Y4, X7, Y7: (U/2)*( X1*sin(B(t,1,3)) + -
Y1*cos(B(t,1,3)) + -X4*sin(B(t,3,4)) + -Y4*cos(B(t,3,4)) + -
X7*sin(B(t,3,7)) + -Y7*cos(B(t,3,7)) )
dY3 = lambda t, X1, Y1, X4, Y4, X7, Y7: (U/2)*( X1*cos(B(t,1,3)) +
Y1*sin(B(t,1,3)) + X4*cos(B(t,3,4)) + -Y4*sin(B(t,3,4))
+ X7*cos(B(t,3,7)) + -Y7*sin(B(t,3,7)) )

```

```

dX4 = lambda t, X2, Y2, X3, Y3, X8, Y8: (U/2)*( X2*sin(B(t,2,4)) + -
Y2*cos(B(t,2,4)) + X3*sin(B(t,3,4)) + -Y3*cos(B(t,3,4)) + -
X8*sin(B(t,4,8)) + -Y8*cos(B(t,4,8)) )
dY4 = lambda t, X2, Y2, X3, Y3, X8, Y8: (U/2)*( X2*cos(B(t,2,4)) +
Y2*sin(B(t,2,4)) + X3*cos(B(t,3,4)) + Y3*sin(B(t,3,4)) +
X8*cos(B(t,4,8)) + -Y8*sin(B(t,4,8)) )

```

```

dX5 = lambda t, X1, Y1, X6, Y6, X7, Y7: (U/2)*( X1*sin(B(t,1,5)) + -
Y1*cos(B(t,1,5)) + -X6*sin(B(t,5,6)) + -Y6*cos(B(t,5,6)) + -
X7*sin(B(t,5,7)) + -Y7*cos(B(t,5,7)) )
dY5 = lambda t, X1, Y1, X6, Y6, X7, Y7: (U/2)*( X1*cos(B(t,1,5)) +
Y1*sin(B(t,1,5)) + X6*cos(B(t,5,6)) + -Y6*sin(B(t,5,6))
+ X7*cos(B(t,5,7)) + -Y7*sin(B(t,5,7)) )

```

```

dX6 = lambda t, X2, Y2, X5, Y5, X8, Y8: (U/2)*( X2*sin(B(t,2,6)) + -
Y2*cos(B(t,2,6)) + X5*sin(B(t,5,6)) + -Y5*cos(B(t,5,6)) + -
X8*sin(B(t,6,8)) + -Y8*cos(B(t,6,8)) )
dY6 = lambda t, X2, Y2, X5, Y5, X8, Y8: (U/2)*( X2*cos(B(t,2,6)) +
Y2*sin(B(t,2,6)) + X5*cos(B(t,5,6)) + Y5*sin(B(t,5,6)) +
X8*cos(B(t,6,8)) + -Y8*sin(B(t,6,8)) )

```

```

dX7 = lambda t, X3, Y3, X5, Y5, X8, Y8: (U/2)*( X3*sin(B(t,3,7)) + -
Y3*cos(B(t,3,7)) + X5*sin(B(t,5,7)) + -Y5*cos(B(t,5,7)) + -
X8*sin(B(t,7,8)) + -Y8*cos(B(t,7,8)) )
dY7 = lambda t, X3, Y3, X5, Y5, X8, Y8: (U/2)*( X3*cos(B(t,3,7)) +
Y3*sin(B(t,3,7)) + X5*cos(B(t,5,7)) + Y5*sin(B(t,5,7)) +
X8*cos(B(t,7,8)) + -Y8*sin(B(t,7,8)) )

```

```

dX8 = lambda t, X4, Y4, X6, Y6, X7, Y7: (U/2)*( X4*sin(B(t,4,8)) + -
Y4*cos(B(t,4,8)) + X6*sin(B(t,6,8)) + -Y6*cos(B(t,6,8))
+ X7*sin(B(t,7,8)) + -Y7*cos(B(t,7,8)) )
dY8 = lambda t, X4, Y4, X6, Y6, X7, Y7: (U/2)*( X4*cos(B(t,4,8)) +
Y4*sin(B(t,4,8)) + X6*cos(B(t,6,8)) + Y6*sin(B(t,6,8)) +
X7*cos(B(t,7,8)) + Y7*sin(B(t,7,8)) )

```

```

# COEFICIENTES K Y L DEL METODO RUNGE-KUTTA CUARTO ORDEN
# Noca: K hace referencia a reales y L a Complejos

```

```

K = np.zeros((5,9))
L = np.zeros((5,9))

```

```

for i in range(0,n-1):

```

```

T[i+1] = T[i] + h

K[1,1] = h*dX1( T[i], vX2[i], vY2[i], vX3[i], vY3[i], vX5[i],
vY5[i] )
K[1,2] = h*dX2( T[i], vX1[i], vY1[i], vX4[i], vY4[i], vX6[i],
vY6[i] )
K[1,3] = h*dX3( T[i], vX1[i], vY1[i], vX4[i], vY4[i], vX7[i],
vY7[i] )
K[1,4] = h*dX4( T[i], vX2[i], vY2[i], vX3[i], vY3[i], vX8[i],
vY8[i] )
K[1,5] = h*dX5( T[i], vX1[i], vY1[i], vX6[i], vY6[i], vX7[i],
vY7[i] )
K[1,6] = h*dX6( T[i], vX2[i], vY2[i], vX5[i], vY5[i], vX8[i],
vY8[i] )
K[1,7] = h*dX7( T[i], vX3[i], vY3[i], vX5[i], vY5[i], vX8[i],
vY8[i] )
K[1,8] = h*dX8( T[i], vX4[i], vY4[i], vX6[i], vY6[i], vX7[i],
vY7[i] )

L[1,1] = h*dY1( T[i], vX2[i], vY2[i], vX3[i], vY3[i], vX5[i],
vY5[i] )
L[1,2] = h*dY2( T[i], vX1[i], vY1[i], vX4[i], vY4[i], vX6[i],
vY6[i] )
L[1,3] = h*dY3( T[i], vX1[i], vY1[i], vX4[i], vY4[i], vX7[i],
vY7[i] )
L[1,4] = h*dY4( T[i], vX2[i], vY2[i], vX3[i], vY3[i], vX8[i],
vY8[i] )
L[1,5] = h*dY5( T[i], vX1[i], vY1[i], vX6[i], vY6[i], vX7[i],
vY7[i] )
L[1,6] = h*dY6( T[i], vX2[i], vY2[i], vX5[i], vY5[i], vX8[i],
vY8[i] )
L[1,7] = h*dY7( T[i], vX3[i], vY3[i], vX5[i], vY5[i], vX8[i],
vY8[i] )
L[1,8] = h*dY8( T[i], vX4[i], vY4[i], vX6[i], vY6[i], vX7[i],
vY7[i] )

K[2,1] = h*dX1( T[i] + h/2, vX2[i] + K[1,2]/2, vY2[i] + L[1,2]/2,
vX3[i] + K[1,3]/2, vY3[i] + L[1,3]/2, vX5[i] + K[1,5]/2, vY5[i] +
L[1,5]/2 )
K[2,2] = h*dX2( T[i] + h/2, vX1[i] + K[1,1]/2, vY1[i] + L[1,1]/2,
vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX6[i] + K[1,6]/2, vY6[i] +
L[1,6]/2 )
K[2,3] = h*dX3( T[i] + h/2, vX1[i] + K[1,1]/2, vY1[i] + L[1,1]/2,
vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX7[i] + K[1,7]/2, vY7[i] +
L[1,7]/2 )
K[2,4] = h*dX4( T[i] + h/2, vX2[i] + K[1,2]/2, vY2[i] + L[1,2]/2,
vX3[i] + K[1,3]/2, vY3[i] + L[1,3]/2, vX8[i] + K[1,8]/2, vY8[i] +
L[1,8]/2 )
K[2,5] = h*dX5( T[i] + h/2, vX1[i] + K[1,1]/2, vY1[i] + L[1,1]/2,
vX6[i] + K[1,6]/2, vY6[i] + L[1,6]/2, vX7[i] + K[1,7]/2, vY7[i] +
L[1,7]/2 )
K[2,6] = h*dX6( T[i] + h/2, vX2[i] + K[1,2]/2, vY2[i] + L[1,2]/2,
vX5[i] + K[1,5]/2, vY5[i] + L[1,5]/2, vX8[i] + K[1,8]/2, vY8[i] +
L[1,8]/2 )

```

$$K[2,7] = h*dX7(T[i] + h/2, vX3[i] + K[1,3]/2, vY3[i] + L[1,3]/2, vX5[i] + K[1,5]/2, vY5[i] + L[1,5]/2, vX8[i] + K[1,8]/2, vY8[i] + L[1,8]/2)$$

$$K[2,8] = h*dX8(T[i] + h/2, vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX6[i] + K[1,6]/2, vY6[i] + L[1,6]/2, vX7[i] + K[1,7]/2, vY7[i] + L[1,7]/2)$$

$$L[2,1] = h*dY1(T[i] + h/2, vX2[i] + K[1,2]/2, vY2[i] + L[1,2]/2, vX3[i] + K[1,3]/2, vY3[i] + L[1,3]/2, vX5[i] + K[1,5]/2, vY5[i] + L[1,5]/2)$$

$$L[2,2] = h*dY2(T[i] + h/2, vX1[i] + K[1,1]/2, vY1[i] + L[1,1]/2, vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX6[i] + K[1,6]/2, vY6[i] + L[1,6]/2)$$

$$L[2,3] = h*dY3(T[i] + h/2, vX1[i] + K[1,1]/2, vY1[i] + L[1,1]/2, vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX7[i] + K[1,7]/2, vY7[i] + L[1,7]/2)$$

$$L[2,4] = h*dY4(T[i] + h/2, vX2[i] + K[1,2]/2, vY2[i] + L[1,2]/2, vX3[i] + K[1,3]/2, vY3[i] + L[1,3]/2, vX8[i] + K[1,8]/2, vY8[i] + L[1,8]/2)$$

$$L[2,5] = h*dY5(T[i] + h/2, vX1[i] + K[1,1]/2, vY1[i] + L[1,1]/2, vX6[i] + K[1,6]/2, vY6[i] + L[1,6]/2, vX7[i] + K[1,7]/2, vY7[i] + L[1,7]/2)$$

$$L[2,6] = h*dY6(T[i] + h/2, vX2[i] + K[1,2]/2, vY2[i] + L[1,2]/2, vX5[i] + K[1,5]/2, vY5[i] + L[1,5]/2, vX8[i] + K[1,8]/2, vY8[i] + L[1,8]/2)$$

$$L[2,7] = h*dY7(T[i] + h/2, vX3[i] + K[1,3]/2, vY3[i] + L[1,3]/2, vX5[i] + K[1,5]/2, vY5[i] + L[1,5]/2, vX8[i] + K[1,8]/2, vY8[i] + L[1,8]/2)$$

$$L[2,8] = h*dY8(T[i] + h/2, vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX6[i] + K[1,6]/2, vY6[i] + L[1,6]/2, vX7[i] + K[1,7]/2, vY7[i] + L[1,7]/2)$$

$$K[3,1] = h*dX1(T[i] + h/2, vX2[i] + K[2,2]/2, vY2[i] + L[2,2]/2, vX3[i] + K[2,3]/2, vY3[i] + L[2,3]/2, vX5[i] + K[2,5]/2, vY5[i] + L[2,5]/2)$$

$$K[3,2] = h*dX2(T[i] + h/2, vX1[i] + K[2,1]/2, vY1[i] + L[2,1]/2, vX4[i] + K[2,4]/2, vY4[i] + L[2,4]/2, vX6[i] + K[2,6]/2, vY6[i] + L[2,6]/2)$$

$$K[3,3] = h*dX3(T[i] + h/2, vX1[i] + K[2,1]/2, vY1[i] + L[2,1]/2, vX4[i] + K[2,4]/2, vY4[i] + L[2,4]/2, vX7[i] + K[2,7]/2, vY7[i] + L[2,7]/2)$$

$$K[3,4] = h*dX4(T[i] + h/2, vX2[i] + K[2,2]/2, vY2[i] + L[2,2]/2, vX3[i] + K[2,3]/2, vY3[i] + L[2,3]/2, vX8[i] + K[2,8]/2, vY8[i] + L[2,8]/2)$$

$$K[3,5] = h*dX5(T[i] + h/2, vX1[i] + K[2,1]/2, vY1[i] + L[2,1]/2, vX6[i] + K[2,6]/2, vY6[i] + L[2,6]/2, vX7[i] + K[2,7]/2, vY7[i] + L[2,7]/2)$$

$$K[3,6] = h*dX6(T[i] + h/2, vX2[i] + K[2,2]/2, vY2[i] + L[2,2]/2, vX5[i] + K[2,5]/2, vY5[i] + L[2,5]/2, vX8[i] + K[2,8]/2, vY8[i] + L[2,8]/2)$$

$$K[3,7] = h*dX7(T[i] + h/2, vX3[i] + K[2,3]/2, vY3[i] + L[2,3]/2, vX5[i] + K[2,5]/2, vY5[i] + L[2,5]/2, vX8[i] + K[2,8]/2, vY8[i] + L[2,8]/2)$$

$$K[3,8] = h * dx8(T[i] + h/2, vx4[i] + K[2,4]/2, vy4[i] + L[2,4]/2, vx6[i] + K[2,6]/2, vy6[i] + L[2,6]/2, vx7[i] + K[2,7]/2, vy7[i] + L[2,7]/2)$$

$$L[3,1] = h * dy1(T[i] + h/2, vx2[i] + K[2,2]/2, vy2[i] + L[2,2]/2, vx3[i] + K[2,3]/2, vy3[i] + L[2,3]/2, vx5[i] + K[2,5]/2, vy5[i] + L[2,5]/2)$$

$$L[3,2] = h * dy2(T[i] + h/2, vx1[i] + K[2,1]/2, vy1[i] + L[2,1]/2, vx4[i] + K[2,4]/2, vy4[i] + L[2,4]/2, vx6[i] + K[2,6]/2, vy6[i] + L[2,6]/2)$$

$$L[3,3] = h * dy3(T[i] + h/2, vx1[i] + K[2,1]/2, vy1[i] + L[2,1]/2, vx4[i] + K[2,4]/2, vy4[i] + L[2,4]/2, vx7[i] + K[2,7]/2, vy7[i] + L[2,7]/2)$$

$$L[3,4] = h * dy4(T[i] + h/2, vx2[i] + K[2,2]/2, vy2[i] + L[2,2]/2, vx3[i] + K[2,3]/2, vy3[i] + L[2,3]/2, vx8[i] + K[2,8]/2, vy8[i] + L[2,8]/2)$$

$$L[3,5] = h * dy5(T[i] + h/2, vx1[i] + K[2,1]/2, vy1[i] + L[2,1]/2, vx6[i] + K[2,6]/2, vy6[i] + L[2,6]/2, vx7[i] + K[2,7]/2, vy7[i] + L[2,7]/2)$$

$$L[3,6] = h * dy6(T[i] + h/2, vx2[i] + K[2,2]/2, vy2[i] + L[2,2]/2, vx5[i] + K[2,5]/2, vy5[i] + L[2,5]/2, vx8[i] + K[2,8]/2, vy8[i] + L[2,8]/2)$$

$$L[3,7] = h * dy7(T[i] + h/2, vx3[i] + K[2,3]/2, vy3[i] + L[2,3]/2, vx5[i] + K[2,5]/2, vy5[i] + L[2,5]/2, vx8[i] + K[2,8]/2, vy8[i] + L[2,8]/2)$$

$$L[3,8] = h * dy8(T[i] + h/2, vx4[i] + K[2,4]/2, vy4[i] + L[2,4]/2, vx6[i] + K[2,6]/2, vy6[i] + L[2,6]/2, vx7[i] + K[2,7]/2, vy7[i] + L[2,7]/2)$$

$$K[4,1] = h * dx1(T[i] + h, vx2[i] + K[3,2], vy2[i] + L[3,2], vx3[i] + K[3,3], vy3[i] + L[3,3], vx5[i] + K[3,5], vy5[i] + L[3,5])$$

$$K[4,2] = h * dx2(T[i] + h, vx1[i] + K[3,1], vy1[i] + L[3,1], vx4[i] + K[3,4], vy4[i] + L[3,4], vx6[i] + K[3,6], vy6[i] + L[3,6])$$

$$K[4,3] = h * dx3(T[i] + h, vx1[i] + K[3,1], vy1[i] + L[3,1], vx4[i] + K[3,4], vy4[i] + L[3,4], vx7[i] + K[3,7], vy7[i] + L[3,7])$$

$$K[4,4] = h * dx4(T[i] + h, vx2[i] + K[3,2], vy2[i] + L[3,2], vx3[i] + K[3,3], vy3[i] + L[3,3], vx8[i] + K[3,8], vy8[i] + L[3,8])$$

$$K[4,5] = h * dx5(T[i] + h, vx1[i] + K[3,1], vy1[i] + L[3,1], vx6[i] + K[3,6], vy6[i] + L[3,6], vx7[i] + K[3,7], vy7[i] + L[3,7])$$

$$K[4,6] = h * dx6(T[i] + h, vx2[i] + K[3,2], vy2[i] + L[3,2], vx5[i] + K[3,5], vy5[i] + L[3,5], vx8[i] + K[3,8], vy8[i] + L[3,8])$$

$$K[4,7] = h * dx7(T[i] + h, vx3[i] + K[3,3], vy3[i] + L[3,3], vx5[i] + K[3,5], vy5[i] + L[3,5], vx8[i] + K[3,8], vy8[i] + L[3,8])$$

$$K[4,8] = h * dx8(T[i] + h, vx4[i] + K[3,4], vy4[i] + L[3,4], vx6[i] + K[3,6], vy6[i] + L[3,6], vx7[i] + K[3,7], vy7[i] + L[3,7])$$

$$L[4,1] = h * dy1(T[i] + h, vx2[i] + K[3,2], vy2[i] + L[3,2], vx3[i] + K[3,3], vy3[i] + L[3,3], vx5[i] + K[3,5], vy5[i] + L[3,5])$$

$$L[4,2] = h * dy2(T[i] + h, vx1[i] + K[3,1], vy1[i] + L[3,1], vx4[i] + K[3,4], vy4[i] + L[3,4], vx6[i] + K[3,6], vy6[i] + L[3,6])$$

$$L[4,3] = h * dy3(T[i] + h, vx1[i] + K[3,1], vy1[i] + L[3,1], vx4[i] + K[3,4], vy4[i] + L[3,4], vx7[i] + K[3,7], vy7[i] + L[3,7])$$

$$L[4,4] = h * dy4(T[i] + h, vx2[i] + K[3,2], vy2[i] + L[3,2], vx3[i] + K[3,3], vy3[i] + L[3,3], vx8[i] + K[3,8], vy8[i] + L[3,8])$$

```

L[4,5] = h*dY5( T[i] + h, vX1[i] + K[3,1], vY1[i] + L[3,1], vX6[i]
+ K[3,6], vY6[i] + L[3,6], vX7[i] + K[3,7], vY7[i] + L[3,7] )
L[4,6] = h*dY6( T[i] + h, vX2[i] + K[3,2], vY2[i] + L[3,2], vX5[i]
+ K[3,5], vY5[i] + L[3,5], vX8[i] + K[3,8], vY8[i] + L[3,8] )
L[4,7] = h*dY7( T[i] + h, vX3[i] + K[3,3], vY3[i] + L[3,3], vX5[i]
+ K[3,5], vY5[i] + L[3,5], vX8[i] + K[3,8], vY8[i] + L[3,8] )
L[4,8] = h*dY8( T[i] + h, vX4[i] + K[3,4], vY4[i] + L[3,4], vX6[i]
+ K[3,6], vY6[i] + L[3,6], vX7[i] + K[3,7], vY7[i] + L[3,7] )

```

```

# SOLUCIONES A ECUACIONES DEL MARCO DE INTERACCION DADAS POR RUNGE-KUTTA
CUARTO ORDEN

```

```

vX1[i+1] = vX1[i] + (1/6)*( K[1,1] + 2*K[2,1] + 2*K[3,1] + K[4,1] )
vY1[i+1] = vY1[i] + (1/6)*( L[1,1] + 2*L[2,1] + 2*L[3,1] + L[4,1] )

vX2[i+1] = vX2[i] + (1/6)*( K[1,2] + 2*K[2,2] + 2*K[3,2] + K[4,2] )
vY2[i+1] = vY2[i] + (1/6)*( L[1,2] + 2*L[2,2] + 2*L[3,2] + L[4,2] )

vX3[i+1] = vX3[i] + (1/6)*( K[1,3] + 2*K[2,3] + 2*K[3,3] + K[4,3] )
vY3[i+1] = vY3[i] + (1/6)*( L[1,3] + 2*L[2,3] + 2*L[3,3] + L[4,3] )

vX4[i+1] = vX4[i] + (1/6)*( K[1,4] + 2*K[2,4] + 2*K[3,4] + K[4,4] )
vY4[i+1] = vY4[i] + (1/6)*( L[1,4] + 2*L[2,4] + 2*L[3,4] + L[4,4] )

vX5[i+1] = vX5[i] + (1/6)*( K[1,5] + 2*K[2,5] + 2*K[3,5] + K[4,5] )
vY5[i+1] = vY5[i] + (1/6)*( L[1,5] + 2*L[2,5] + 2*L[3,5] + L[4,5] )

vX6[i+1] = vX6[i] + (1/6)*( K[1,6] + 2*K[2,6] + 2*K[3,6] + K[4,6] )
vY6[i+1] = vY6[i] + (1/6)*( L[1,6] + 2*L[2,6] + 2*L[3,6] + L[4,6] )

vX7[i+1] = vX7[i] + (1/6)*( K[1,7] + 2*K[2,7] + 2*K[3,7] + K[4,7] )
vY7[i+1] = vY7[i] + (1/6)*( L[1,7] + 2*L[2,7] + 2*L[3,7] + L[4,7] )

vX8[i+1] = vX8[i] + (1/6)*( K[1,8] + 2*K[2,8] + 2*K[3,8] + K[4,8] )
vY8[i+1] = vY8[i] + (1/6)*( L[1,8] + 2*L[2,8] + 2*L[3,8] + L[4,8] )

```

```

# SOLUCIONES A ECUACIONES DE LA FUNCION DE ONDA DADAS POR RUNGE-KUTTA
CUARTO ORDEN

```

```

cX1[i] = vX1[i]*cos(E[0]*T[i]) - vY1[i]*sin(E[0]*T[i])
cY1[i] = vX1[i]*sin(E[0]*T[i]) + vY1[i]*cos(E[0]*T[i])

cX2[i] = vX2[i]*cos(E[1]*T[i]) - vY2[i]*sin(E[1]*T[i])
cY2[i] = vX2[i]*sin(E[1]*T[i]) + vY2[i]*cos(E[1]*T[i])

cX3[i] = vX3[i]*cos(E[2]*T[i]) - vY3[i]*sin(E[2]*T[i])
cY3[i] = vX3[i]*sin(E[2]*T[i]) + vY3[i]*cos(E[2]*T[i])

cX4[i] = vX4[i]*cos(E[3]*T[i]) - vY4[i]*sin(E[3]*T[i])
cY4[i] = vX4[i]*sin(E[3]*T[i]) + vY4[i]*cos(E[3]*T[i])

cX5[i] = vX5[i]*cos(E[4]*T[i]) - vY5[i]*sin(E[4]*T[i])
cY5[i] = vX5[i]*sin(E[4]*T[i]) + vY5[i]*cos(E[4]*T[i])

cX6[i] = vX6[i]*cos(E[5]*T[i]) - vY6[i]*sin(E[5]*T[i])
cY6[i] = vX6[i]*sin(E[5]*T[i]) + vY6[i]*cos(E[5]*T[i])

```

$$\begin{aligned} cX7[i] &= vX7[i]*\cos(E[6]*T[i]) - vY7[i]*\sin(E[6]*T[i]) \\ cY7[i] &= vX7[i]*\sin(E[6]*T[i]) + vY7[i]*\cos(E[6]*T[i]) \end{aligned}$$

$$\begin{aligned} cX8[i] &= vX8[i]*\cos(E[7]*T[i]) - vY8[i]*\sin(E[7]*T[i]) \\ cY8[i] &= vX8[i]*\sin(E[7]*T[i]) + vY8[i]*\cos(E[7]*T[i]) \end{aligned}$$

CALCULO DE PROBABILIDADES POR CADA ESTADO

$$\begin{aligned} D1[i] &= (cX1[i])**2 + (cY1[i])**2 \\ D2[i] &= (cX2[i])**2 + (cY2[i])**2 \\ D3[i] &= (cX3[i])**2 + (cY3[i])**2 \\ D4[i] &= (cX4[i])**2 + (cY4[i])**2 \\ D5[i] &= (cX5[i])**2 + (cY5[i])**2 \\ D6[i] &= (cX6[i])**2 + (cY6[i])**2 \\ D7[i] &= (cX7[i])**2 + (cY7[i])**2 \\ D8[i] &= (cX8[i])**2 + (cY8[i])**2 \end{aligned}$$

$$N[i] = D1[i] + D2[i] + D3[i] + D4[i] + D5[i] + D6[i] + D7[i] + D8[i]$$

$$\begin{aligned} cX1[-1] &= vX1[-1]*\cos(E[0]*T[-1]) - vY1[-1]*\sin(E[0]*T[-1]) \\ cY1[-1] &= vX1[-1]*\sin(E[0]*T[-1]) + vY1[-1]*\cos(E[0]*T[-1]) \end{aligned}$$

$$\begin{aligned} cX2[-1] &= vX2[-1]*\cos(E[1]*T[-1]) - vY2[-1]*\sin(E[1]*T[-1]) \\ cY2[-1] &= vX2[-1]*\sin(E[1]*T[-1]) + vY2[-1]*\cos(E[1]*T[-1]) \end{aligned}$$

$$\begin{aligned} cX3[-1] &= vX3[-1]*\cos(E[2]*T[-1]) - vY3[-1]*\sin(E[2]*T[-1]) \\ cY3[-1] &= vX3[-1]*\sin(E[2]*T[-1]) + vY3[-1]*\cos(E[2]*T[-1]) \end{aligned}$$

$$\begin{aligned} cX4[-1] &= vX4[-1]*\cos(E[3]*T[-1]) - vY4[-1]*\sin(E[3]*T[-1]) \\ cY4[-1] &= vX4[-1]*\sin(E[3]*T[-1]) + vY4[-1]*\cos(E[3]*T[-1]) \end{aligned}$$

$$\begin{aligned} cX5[-1] &= vX5[-1]*\cos(E[4]*T[-1]) - vY5[-1]*\sin(E[4]*T[-1]) \\ cY5[-1] &= vX5[-1]*\sin(E[4]*T[-1]) + vY5[-1]*\cos(E[4]*T[-1]) \end{aligned}$$

$$\begin{aligned} cX6[-1] &= vX6[-1]*\cos(E[5]*T[-1]) - vY6[-1]*\sin(E[5]*T[-1]) \\ cY6[-1] &= vX6[-1]*\sin(E[5]*T[-1]) + vY6[-1]*\cos(E[5]*T[-1]) \end{aligned}$$

$$\begin{aligned} cX7[-1] &= vX7[-1]*\cos(E[6]*T[-1]) - vY7[-1]*\sin(E[6]*T[-1]) \\ cY7[-1] &= vX7[-1]*\sin(E[6]*T[-1]) + vY7[-1]*\cos(E[6]*T[-1]) \end{aligned}$$

$$\begin{aligned} cX8[-1] &= vX8[-1]*\cos(E[7]*T[-1]) - vY8[-1]*\sin(E[7]*T[-1]) \\ cY8[-1] &= vX8[-1]*\sin(E[7]*T[-1]) + vY8[-1]*\cos(E[7]*T[-1]) \end{aligned}$$

$$\begin{aligned} D1[-1] &= (cX1[-1])**2 + (cY1[-1])**2 \\ D2[-1] &= (cX2[-1])**2 + (cY2[-1])**2 \\ D3[-1] &= (cX3[-1])**2 + (cY3[-1])**2 \\ D4[-1] &= (cX4[-1])**2 + (cY4[-1])**2 \\ D5[-1] &= (cX5[-1])**2 + (cY5[-1])**2 \\ D6[-1] &= (cX6[-1])**2 + (cY6[-1])**2 \\ D7[-1] &= (cX7[-1])**2 + (cY7[-1])**2 \\ D8[-1] &= (cX8[-1])**2 + (cY8[-1])**2 \end{aligned}$$

$$N[-1] = D1[-1] + D2[-1] + D3[-1] + D4[-1] + D5[-1] + D6[-1] + D7[-1] + D8[-1]$$

```

XMAX = np.array([max(cX1), max(cX2), max(cX3), max(cX4), max(cX5),
max(cX6), max(cX7), max(cX8)])
YMAX = np.array([max(cY1), max(cY2), max(cY3), max(cY4), max(cY5),
max(cY6), max(cY7), max(cY8)])
TMAX = max( np.concatenate( (XMAX,YMAX), axis = None ) ) + 0.1

XMIN = np.array([min(cX1), min(cX2), min(cX3), min(cX4), min(cX5),
min(cX6), min(cX7), min(cX8)])
YMIN = np.array([min(cY1), min(cY2), min(cY3), min(cY4), min(cY5),
min(cY6), min(cY7), min(cY8)])
TMIN = min( np.concatenate( (XMIN,YMIN), axis = None ) ) - 0.1

# GRAFICAS DE LA SOLUCION Y COEFICIENTES COMPLEJOS DE FUNCION DE ONDA.

sstyle = "seaborn-poster"
plt.style.use(sstyle)
plt.rc('text', usetex = True)
plt.rc('font', family = 'serif')

#""""
plt.figure(figsize = (20,10))

file_name = " X1_" + "%.3f"%vX1[0] + " Y1_" + "%.3f"%vY1[0] + "
X2_" + "%.3f"%vX2[0] + " Y2_" + "%.3f"%vY2[0] \
+ " X3_" + "%.3f"%vX3[0] + " Y3_" + "%.3f"%vY3[0] + "
X4_" + "%.3f"%vX4[0] + " Y4_" + "%.3f"%vY4[0] \
+ " X5_" + "%.3f"%vX5[0] + " Y5_" + "%.3f"%vY5[0] + "
X6_" + "%.3f"%vX6[0] + " Y6_" + "%.3f"%vY6[0] \
+ " X7_" + "%.3f"%vX7[0] + " Y7_" + "%.3f"%vY7[0] + "
X8_" + "%.3f"%vX8[0] + " Y8_" + "%.3f"%vY8[0] \
+ " w1_" + "%.3f"%w1 + " w2_" + "%.3f"%w2 + "
w3_" + "%.3f"%w3 + " J_" + "%.3f"%J \
+ " Jp_" + "%.3f"%Jp + " Ome_" + "%.3f"%U + "
ome_" + "%.3f"%w + " omega0_" + "%.3f"%w0 \
+ " phi_" + "%.3f"%phi + " h_" + "%.3f"%h + "_"

plt.plot(T, N, label = "Total Probability", linewidth = 5.0 )
plt.plot(T, D1, label = "$|C_{000}(t)|^2$")
plt.plot(T, D2, label = "$|C_{001}(t)|^2$")
plt.plot(T, D3, label = "$|C_{010}(t)|^2$")
plt.plot(T, D4, label = "$|C_{011}(t)|^2$")
plt.plot(T, D5, label = "$|C_{100}(t)|^2$")
plt.plot(T, D6, label = "$|C_{101}(t)|^2$")
plt.plot(T, D7, label = "$|C_{110}(t)|^2$")
plt.plot(T, D8, label = "$|C_{111}(t)|^2$")

plt.title( "Three-Qubit Probabilities")
plt.xlabel("Time (s)")
plt.ylabel("Probability")

plt.legend(loc=0)
plt.grid()

#plt.savefig( file_name + ".png", format = "png")
plt.show()

```

```

# COEFICIENTES DE LA FUNCION DE ONDA

COMP = input( "Do you want to see the components of the wave function?
[Y/N]: ")

if(COMP == 'N' or COMP == 'n' or COMP == 'No' or COMP == 'NO' or COMP ==
'no' or COMP == 'Nope' or COMP == 'nope'):
    pass

elif (COMP == 'Y' or COMP == 'y' or COMP == 'Yes' or COMP == 'YES' or
COMP == 'yes'):

# -----
# PRIMER COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX1, label = "$X_{1}$ - Real ")
plt.plot(T, cY1, label = "$Y_{1}$ - Imaginary")
plt.title( "Components of the Coefficient $C_{000}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEGUNDO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX2, label = "$X_{2}$ - Real ")
plt.plot(T, cY2, label = "$Y_{2}$ - Imaginary")
plt.title( "Components of the Coefficient $C_{001}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# TERCER COEFICIENTE
# -----

plt.figure(figsize = (20,10))

```

```

plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX3, label = "$X_{3}$ - Real ")
plt.plot(T, cY3, label = "$Y_{3}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{010}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# CUARTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX4, label = "$X_{4}$ - Real ")
plt.plot(T, cY4, label = "$Y_{4}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{011}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# QUINTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX5, label = "$X_{5}$ - Real ")
plt.plot(T, cY5, label = "$Y_{5}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{100}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEXTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX6, label = "$X_{6}$ - Real ")
plt.plot(T, cY6, label = "$Y_{6}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{101}(t)$")

```

```

plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEPTIMO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX7, label = "$X_{7}$ - Real ")
plt.plot(T, cY7, label = "$Y_{7}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{110}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# OCTAVO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX8, label = "$X_{8}$ - Real ")
plt.plot(T, cY8, label = "$Y_{8}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{111}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

```

Anexo 4. Código en PYTHON para graficar los resultados numéricos de FORTRAN

```
"""
PROGRAMA PARA GRAFICAR SOLUCIONES NUMERICAS
DEPENDIENTES DE UN ARCHIVO TXT - PAP4G03
11/10/2018 - 16/11/2018
PLOT.py
"""

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import colors as mcolors
from matplotlib import rc
import pandas as pd
import os

def Plot1Qubit(RUN, COMP):

    # EXPORTACION DE LOS DATOS GENERADOS POR FORTRAN

    figpath = "/F95Fig/1QFig/"

    file_name = "RelQ_R" + str(RUN)
    df = pd.read_csv(file_name + ".txt", sep = " ", header = None,
skipinitialspace = True)
    df = df.drop(8,1)
    df.columns = ["Time", "X0", "Y0", "X1", "Y1", "C0", "C1", "Norm"]

    XMAX = np.array( [ max(df["X0"]), max(df["X1"]) ] )
    YMAX = np.array( [ max(df["Y0"]), max(df["Y1"]) ] )
    TMAX = max( np.concatenate( (XMAX,YMAX), axis = None ) ) + 0.1

    XMIN = np.array( [ min(df["X0"]), min(df["X1"]) ] )
    YMIN = np.array( [ min(df["Y0"]), min(df["Y1"]) ] )
    TMIN = min( np.concatenate( (XMIN,YMIN), axis = None ) ) - 0.1

    # GRAFICAS DE LOS DATOS

    sstyle = "seaborn-poster"
    plt.style.use(sstyle)
    plt.rc('text', usetex=True)
    plt.rc('font', family='serif')

    plt.figure(figsize = (20,10))

    plt.plot(df["Time"], df["Norm"], label = "Total Probability")
    plt.plot(df["Time"], df["C0"], label = "$|C_{0}(t)|^2$")
    plt.plot(df["Time"], df["C1"], label = "$|C_{1}(t)|^2$")

    plt.title("One-Qubit Probabilities")
    plt.xlabel("Time (s)")
    plt.ylabel("Probability")

    plt.legend()
```

```

plt.grid()
plt.show()

#plt.savefig( figpath + file_name + ".png", format = "png")
if(COMP == 'N' or COMP == 'n' or COMP == 'No' or COMP == 'NO' or COMP
== 'no' or COMP == 'Nope' or COMP == 'nope'):
    pass

elif (COMP == 'Y' or COMP == 'y' or COMP == 'Yes' or COMP == 'YES' or
COMP == 'yes'):

    # GRAFICA DE LOS COEFICIENTES DE LA FUNCION DE ONDA DEL SISTEMA
    # -----
    # PRIMER COEFICIENTE
    # -----

    plt.figure(figsize = (20,10))
    plt.ylim(top = TMAX)
    plt.ylim(bottom = TMIN)

    plt.plot(df["Time"], df["X0"], label = "$X_{0}$ - Real ")
    plt.plot(df["Time"], df["Y0"],label = "$Y_{0}$ - Imaginary")
    plt.title( "Components of the Coeficient $C_{0}(t)$")
    plt.xlabel("Time (s)")
    plt.ylabel(" ")
    plt.legend(loc=0)
    plt.grid()
    plt.show()

    # -----
    # SEGUNDO COEFICIENTE
    # -----

    plt.figure(figsize = (20,10))
    plt.ylim(top = TMAX)
    plt.ylim(bottom = TMIN)

    plt.plot(df["Time"], df["X1"], label = "$X_{1}$ - Real ")
    plt.plot(df["Time"], df["Y1"], label = "$Y_{1}$ - Imaginary")
    plt.title( "Components of the Coeficient $C_{1}(t)$")
    plt.xlabel("Time (s)")
    plt.ylabel(" ")
    plt.legend(loc=0)
    plt.grid()
    plt.show()

return

def Plot2Qubit (RUN,COMP):

    # EXPORTACION DE LOS DATOS GENERADOS POR FORTRAN

```

```

figpath = "/F95Fig/2QFig/"

file_name = "Re2Q_R" + str(RUN)
df = pd.read_csv(file_name + ".txt", sep = " ", header = None,
skipinitialspace = True)
df = df.drop(14,1)
df.columns = ["Time", "X00", "Y00", "X01", "Y01", "X10", "Y10",
"X11", "Y11", "C00", "C01", "C10", "C11", "Norm"]

XMAX = np.array( [ max(df["X00"]), max(df["X01"]), max(df["X10"]),
max(df["X11"]) ] )
YMAX = np.array( [ max(df["Y00"]), max(df["Y01"]), max(df["Y10"]),
max(df["Y11"]) ] )
TMAX = max( np.concatenate( (XMAX,YMAX), axis = None ) ) + 0.1

XMIN = np.array( [ min(df["X00"]), min(df["X01"]), min(df["X10"]),
min(df["X11"]) ] )
YMIN = np.array( [ min(df["Y00"]), min(df["Y01"]), min(df["Y10"]),
min(df["Y11"]) ] )
TMIN = min( np.concatenate( (XMIN,YMIN), axis = None ) ) - 0.1

# GRAFICAS DE LOS DATOS

sstyle = "seaborn-poster"
plt.style.use(sstyle)
plt.rc('text', usetex=True)
plt.rc('font', family='serif')

plt.figure(figsize = (20,10))

plt.plot(df["Time"], df["Norm"], label = "Total Probability")
plt.plot(df["Time"], df["C00"] , label = "$|C_{00}(t)|^2$")
plt.plot(df["Time"], df["C01"] , label = "$|C_{01}(t)|^2$")
plt.plot(df["Time"], df["C10"] , label = "$|C_{10}(t)|^2$")
plt.plot(df["Time"], df["C11"] , label = "$|C_{11}(t)|^2$")

plt.title("Two-Qubit Probabilities")
plt.xlabel("Time (s)")
plt.ylabel("Probability")

plt.legend()
plt.grid()
plt.show()

#plt.savefig( figpath + file_name + ".png", format = "png")
if(COMP == 'N' or COMP == 'n' or COMP == 'No' or COMP == 'NO' or COMP
== 'no' or COMP == 'Nope' or COMP == 'nope'):
    pass

elif (COMP == 'Y' or COMP == 'y' or COMP == 'Yes' or COMP == 'YES' or
COMP == 'yes'):

    # GRAFICA DE LOS COEFICIENTES DE LA FUNCION DE ONDA DEL SISTEMA
    # -----
    # PRIMER COEFICIENTE
    # -----

```

```

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X00"], label = "$X_{00}$ - Real ")
plt.plot(df["Time"], df["Y00"], label = "$Y_{00}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{00}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEGUNDO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X01"], label = "$X_{01}$ - Real ")
plt.plot(df["Time"], df["Y01"], label = "$Y_{01}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{01}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# TERCER COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X10"], label = "$X_{10}$ - Real ")
plt.plot(df["Time"], df["Y10"], label = "$Y_{10}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{10}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# CUARTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X11"], label = "$X_{11}$ - Real ")

```

```

plt.plot(df["Time"], df["Y11"], label = "$Y_{11}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{11}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

return

def Plot3Qubit(RUN,COMP):

    # EXPORTACION DE LOS DATOS GENERADOS POR FORTRAN

    figpath = "/F95Fig/3QFig/"

    file_name = "Re3Q_R" + str(RUN)
    df = pd.read_csv(file_name + ".txt", sep = " ", header = None,
skipinitialspace = True)
    df = df.drop(26,1)
    df.columns = ["Time", "X1", "Y1", "X2", "Y2", "X3", "Y3", "X4", "Y4",
                  "X5", "Y5", "X6", "Y6", "X7", "Y7", "X8", "Y8",
"C1",
                  "C2", "C3", "C4", "C5", "C6", "C7", "C8", "Norm"]

    XMAX = np.array( [max(df["X1"]), max(df["X2"]), max(df["X3"]),
max(df["X4"]), max(df["X5"]), max(df["X6"]), max(df["X7"]), max(df["X8"])
])
    YMAX = np.array( [max(df["Y1"]), max(df["Y2"]), max(df["Y3"]),
max(df["Y4"]), max(df["Y5"]), max(df["Y6"]), max(df["Y7"]), max(df["Y8"])
])
    TMAX = max( np.concatenate( (XMAX,YMAX), axis = None ) ) + 0.1

    XMIN = np.array( [min(df["X1"]), min(df["X2"]), min(df["X3"]),
min(df["X4"]), min(df["X5"]), min(df["X6"]), min(df["X7"]), min(df["X8"])
])
    YMIN = np.array( [min(df["Y1"]), min(df["Y2"]), min(df["Y3"]),
min(df["Y4"]), min(df["Y5"]), min(df["Y6"]), min(df["Y7"]), min(df["Y8"])
])
    TMIN = min( np.concatenate( (XMIN,YMIN), axis = None ) ) - 0.1

    # GRAFICAS DE LOS DATOS

    sstyle = "seaborn-poster"
    plt.style.use(sstyle)
    plt.rc('text', usetex=True)
    plt.rc('font', family='serif')
    plt.figure(figsize = (20,10))

    plt.plot( df["Time"], df["Norm"], label = "Total Probability")
    plt.plot( df["Time"], df["C1"] , label = "$|C_{000}(t)|^2$")
    plt.plot( df["Time"], df["C2"] , label = "$|C_{001}(t)|^2$")
    plt.plot( df["Time"], df["C3"] , label = "$|C_{010}(t)|^2$")
    plt.plot( df["Time"], df["C4"] , label = "$|C_{011}(t)|^2$")
    plt.plot( df["Time"], df["C5"] , label = "$|C_{100}(t)|^2$")
    plt.plot( df["Time"], df["C6"] , label = "$|C_{101}(t)|^2$")

```

```

plt.plot( df["Time"], df["C7"] , label = "$|C_{110}(t)|^2$")
plt.plot( df["Time"], df["C8"] , label = "$|C_{111}(t)|^2$")

plt.title("Three-Qubit Probabilities")
plt.xlabel("Time (s)")
plt.ylabel("Probability")

plt.legend()
plt.grid()
plt.show()

#plt.savefig( figpath + file_name + ".png", format = "png")
if(COMP == 'N' or COMP == 'n' or COMP == 'No' or COMP == 'NO' or COMP
== 'no' or COMP == 'Nope' or COMP == 'nope'):
    pass

elif (COMP == 'Y' or COMP == 'y' or COMP == 'Yes' or COMP == 'YES' or
COMP == 'yes'):

    # GRAFICA DE LOS COEFICIENTES DE LA FUNCION DE ONDA DEL SISTEMA
    # -----
    # PRIMER COEFICIENTE
    # -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X1"], label = "$X_{1}$ - Real ")
plt.plot(df["Time"], df["Y1"], label = "$Y_{1}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{000}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEGUNDO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X2"], label = "$X_{2}$ - Real ")
plt.plot(df["Time"], df["Y2"], label = "$Y_{2}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{001}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

```

```

# -----
# TERCER COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X3"], label = "$X_{3}$ - Real ")
plt.plot(df["Time"], df["Y3"], label = "$Y_{3}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{010}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# CUARTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X4"], label = "$X_{4}$ - Real ")
plt.plot(df["Time"], df["Y4"], label = "$Y_{4}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{011}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# QUINTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X5"], label = "$X_{5}$ - Real ")
plt.plot(df["Time"], df["Y5"], label = "$Y_{5}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{100}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEXTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)

```

```

plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X6"], label = "$X_{6}$ - Real ")
plt.plot(df["Time"], df["Y6"], label = "$Y_{6}$ - Imaginary")
plt.title( "Components of the Coeficient  $\bar{C}_{101}(t)$ ")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEPTIMO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X7"], label = "$X_{7}$ - Real ")
plt.plot(df["Time"], df["Y7"], label = "$Y_{7}$ - Imaginary")
plt.title( "Components of the Coeficient  $\bar{C}_{110}(t)$ ")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# OCTAVO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X8"], label = "$X_{8}$ - Real ")
plt.plot(df["Time"], df["Y8"], label = "$Y_{8}$ - Imaginary")
plt.title( "Components of the Coeficient  $\bar{C}_{111}(t)$ ")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

return

def QTeleportation(RUN,COMP):

# EXPORTACION DE LOS DATOS GENERADOS POR FORTRAN

figpath = "/F95Fig/3QFig/"

file_name = "ReQT_R" + str(RUN)

```

```

df = pd.read_csv(file_name + ".txt", sep = " ", header = None,
skipinitialspace = True)
df = df.drop(26,1)
df.columns = ["Time", "X1", "Y1", "X2", "Y2", "X3", "Y3", "X4", "Y4",
              "X5", "Y5", "X6", "Y6", "X7", "Y7", "X8", "Y8",
"C1",
              "C2", "C3", "C4", "C5", "C6", "C7", "C8", "Norm"]

XMAX = np.array( [max(df["X1"]), max(df["X2"]), max(df["X3"]),
max(df["X4"]), max(df["X5"]), max(df["X6"]), max(df["X7"]), max(df["X8"])
])
YMAX = np.array( [max(df["Y1"]), max(df["Y2"]), max(df["Y3"]),
max(df["Y4"]), max(df["Y5"]), max(df["Y6"]), max(df["Y7"]), max(df["Y8"])
])
TMAX = max( np.concatenate( (XMAX,YMAX), axis = None ) ) + 0.1

XMIN = np.array( [min(df["X1"]), min(df["X2"]), min(df["X3"]),
min(df["X4"]), min(df["X5"]), min(df["X6"]), min(df["X7"]), min(df["X8"])
])
YMIN = np.array( [min(df["Y1"]), min(df["Y2"]), min(df["Y3"]),
min(df["Y4"]), min(df["Y5"]), min(df["Y6"]), min(df["Y7"]), min(df["Y8"])
])
TMIN = min( np.concatenate( (XMIN,YMIN), axis = None ) ) - 0.1

# GRAFICAS DE LOS DATOS

sstyle = "seaborn-poster"
plt.style.use(sstyle)
plt.rc('text', usetex=True)
plt.rc('font', family='serif')
plt.figure(figsize = (20,10))

plt.plot( df["Time"], df["Norm"], label = "Total Probability")
plt.plot( df["Time"], df["C8"] , label = "$|C_{111}\rangle(t)|^2$")
plt.plot( df["Time"], df["C7"] , label = "$|C_{110}\rangle(t)|^2$")
plt.plot( df["Time"], df["C6"] , label = "$|C_{101}\rangle(t)|^2$")
plt.plot( df["Time"], df["C5"] , label = "$|C_{100}\rangle(t)|^2$")
plt.plot( df["Time"], df["C4"] , label = "$|C_{011}\rangle(t)|^2$")
plt.plot( df["Time"], df["C3"] , label = "$|C_{010}\rangle(t)|^2$")
plt.plot( df["Time"], df["C2"] , label = "$|C_{001}\rangle(t)|^2$")
plt.plot( df["Time"], df["C1"] , label = "$|C_{000}\rangle(t)|^2$")

plt.title("Three-Qubit Probabilities of Quantum Teleportation")
plt.xlabel("Time (s)")
plt.ylabel("Probability")

plt.legend()
plt.grid()
plt.show()

#plt.savefig( figpath + file_name + ".png", format = "png")
if(COMP == 'N' or COMP == 'n' or COMP == 'No' or COMP == 'NO' or COMP
== 'no' or COMP == 'Nope' or COMP == 'nope'):
    pass

elif (COMP == 'Y' or COMP == 'y' or COMP == 'Yes' or COMP == 'YES' or
COMP == 'yes'):

```

```

# GRAFICA DE LOS COEFICIENTES DE LA FUNCION DE ONDA DEL SISTEMA
# -----
# PRIMER COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X1"], label = "$X_{1}$ - Real ")
plt.plot(df["Time"], df["Y1"], label = "$Y_{1}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{000}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEGUNDO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X2"], label = "$X_{2}$ - Real ")
plt.plot(df["Time"], df["Y2"], label = "$Y_{2}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{001}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# TERCER COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X3"], label = "$X_{3}$ - Real ")
plt.plot(df["Time"], df["Y3"], label = "$Y_{3}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{010}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----

```

```

# CUARTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X4"], label = "$X_{4}$ - Real ")
plt.plot(df["Time"], df["Y4"], label = "$Y_{4}$ - Imaginary")
plt.title( "Components of the Coeficiente $C_{011}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# QUINTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X5"], label = "$X_{5}$ - Real ")
plt.plot(df["Time"], df["Y5"], label = "$Y_{5}$ - Imaginary")
plt.title( "Components of the Coeficiente $C_{100}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEXTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X6"], label = "$X_{6}$ - Real ")
plt.plot(df["Time"], df["Y6"], label = "$Y_{6}$ - Imaginary")
plt.title( "Components of the Coeficiente $C_{101}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEPTIMO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

```

```

plt.plot(df["Time"], df["X7"], label = "$X_{7}$ - Real ")
plt.plot(df["Time"], df["Y7"], label = "$Y_{7}$ - Imaginary")
plt.title( "Components of the Coeficiente $C_{110}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# OCTAVO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(df["Time"], df["X8"], label = "$X_{8}$ - Real ")
plt.plot(df["Time"], df["Y8"], label = "$Y_{8}$ - Imaginary")
plt.title( "Components of the Coeficiente $C_{111}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

return

qubit = int(input("What do you want to plot:\n" +
    "[1] 1-Qubit \n" + "[2] 2-Qubits \n" + "[3] 3-Qubits \n" + "[4]
Quantum Teleportation \n" +
    "Option: "))

run = int(input("Run of file: "))

comp = input( "Do you want to see the components of the wave function?
[Y/N]: ")

if (qubit == 1):
    Plot1Qubit(run,comp)

elif (qubit == 2):
    Plot2Qubit(run,comp)

elif (qubit == 3):
    Plot3Qubit(run,comp)

elif (plot == 4):
    QTeleportation(run,comp)

```

Anexo 5. Código en PYTHON para generar gráficas de Estados de Bell

```
"""
PROGRAMA QUE RESUELVE NUMERICAMENTE LAS FUNCIONES D_00(t), D_01(t),
D_10(t) y D_11(t) QUE SE OBTIENEN AL TRATAR LA ECUACION DEL ESQUEMA DE
INTERACCIONN PARA DOS QUBITS DE CARBONO C13 EN UN ENLACE DE DIAMANTE BAJO
LA INFLUENCIA DE UN CAMPO MAGNETICO PARA LA GENERACION DE ESTADOS DE BELL.

04/10/2018 - 15/11/2018
EBell.py
"""

import numpy as np
from numpy import sin, cos, sqrt, pi
import matplotlib.pyplot as plt
from matplotlib.pyplot import savefig
import matplotlib.path_effects as path_effects

print("*****")
print("PURE BELL-STATE GRAPHS")
print("*****")
bell = int(input("Choose the initial state: \n" + "[1] - 00 \n" + "[2] -
01\n" + "[3] - 10\n" + "[4] - 11\n" + "Option: "))
COMP = input( "Do you want to see the components of the wave function?
[Y/N]: ")

#Constantes del sistema
w0 = 1/pi

U = 0.1/w0
w1 = 100.0/w0
w2 = 200.0/w0
J = 5.0/w0

phi = 0.000

pp = pi/U #PI PULSO
hpp = pi/(2*U) # PI MEDIO PULSO

ti = 0 #Siempre cero

tf1 = hpp
tf2 = tf1 + pp
tf = tf2 + hpp

h = 0.001
n = int((tf-ti)/h)

# Energias del sistema por unidad de h-barra

E = lambda a2,a1: (-1/2)*( (0.5*J)*((-1)**(a1+a2)) + w1*(-1)**a1 + w2*(-
1)**a2 )

E00 = E(0,0)
E01 = E(0,1)
E10 = E(1,0)
E11 = E(1,1)

FO-DGA-CPAP-0017
```

```

w = 0

# Arreglos para solucion numerica
vX00, vY00 = np.zeros(n), np.zeros(n)
vX01, vY01 = np.zeros(n), np.zeros(n)
vX10, vY10 = np.zeros(n), np.zeros(n)
vX11, vY11 = np.zeros(n), np.zeros(n)

cX00, cY00 = np.zeros(n), np.zeros(n)
cX01, cY01 = np.zeros(n), np.zeros(n)
cX10, cY10 = np.zeros(n), np.zeros(n)
cX11, cY11 = np.zeros(n), np.zeros(n)

D00 = np.zeros(n)
D01 = np.zeros(n)
D10 = np.zeros(n)
D11 = np.zeros(n)

N = np.zeros(n)
T = np.zeros(n)
T[0] = ti

# Condiciones inidicales del esquema de interaccion

if (bell == 1):
    vX00[0] = 1

elif(bell == 2):
    vX01[0] = 1

elif(bell == 3):
    vX10[0] = 1

elif(bell == 4):
    vX11[0] = 1

# Condiciones inidicales de la funcion de onda
cX00[0] = vX00[0]*cos(E00*T[0]) - vY00[0]*sin(E00*T[0])
cY00[0] = vX00[0]*sin(E00*T[0]) + vY00[0]*cos(E00*T[0])

cX01[0] = vX01[0]*cos(E01*T[0]) - vY01[0]*sin(E01*T[0])
cY01[0] = vX01[0]*sin(E01*T[0]) + vY01[0]*cos(E01*T[0])

cX10[0] = vX10[0]*cos(E10*T[0]) - vY10[0]*sin(E10*T[0])
cY10[0] = vX10[0]*sin(E10*T[0]) + vY10[0]*cos(E10*T[0])

cX11[0] = vX11[0]*cos(E11*T[0]) - vY11[0]*sin(E11*T[0])
cY11[0] = vX11[0]*sin(E11*T[0]) + vY11[0]*cos(E11*T[0])

# angulos que describen el sistema
theta = lambda t: w*t + phi
ooA01 = lambda t: -(w1+(J/2))*t + theta(t)
ooA10 = lambda t: -(w2+(J/2))*t + theta(t)
iiA01 = lambda t: (w2-(J/2))*t - theta(t)
iiA10 = lambda t: (w1-(J/2))*t - theta(t)

```

```

# Ecuaciones diferenciales que describen el sistema.
dX00 = lambda t, X01, X10, Y01, Y10: (U/2)*(- X01*np.sin(ooA01(t)) -
Y01*np.cos(ooA01(t)) - X10*np.sin(ooA10(t)) - Y10*np.cos(ooA10(t)) )
dY00 = lambda t, X01, X10, Y01, Y10: (U/2)*( X01*np.cos(ooA01(t)) -
Y01*np.sin(ooA01(t)) + X10*np.cos(ooA10(t)) - Y10*np.sin(ooA10(t)) )

dX11 = lambda t, X01, X10, Y01, Y10: (U/2)*(- X01*np.sin(iiA01(t)) -
Y01*np.cos(iiA01(t)) - X10*np.sin(iiA10(t)) - Y10*np.cos(iiA10(t)) )
dY11 = lambda t, X01, X10, Y01, Y10: (U/2)*( X01*np.cos(iiA01(t)) -
Y01*np.sin(iiA01(t)) + X10*np.cos(iiA10(t)) - Y10*np.sin(iiA10(t)) )

dX01 = lambda t, X00, X11, Y00, Y11: (U/2)*( X00*np.sin(ooA01(t)) -
Y00*np.cos(ooA01(t)) + X11*np.sin(iiA01(t)) - Y11*np.cos(iiA01(t)) )
dY01 = lambda t, X00, X11, Y00, Y11: (U/2)*( X00*np.cos(ooA01(t)) +
Y00*np.sin(ooA01(t)) + X11*np.cos(iiA01(t)) + Y11*np.sin(iiA01(t)) )

dX10 = lambda t, X00, X11, Y00, Y11: (U/2)*( X00*np.sin(ooA10(t)) -
Y00*np.cos(ooA10(t)) + X11*np.sin(iiA10(t)) - Y11*np.cos(iiA10(t)) )
dY10 = lambda t, X00, X11, Y00, Y11: (U/2)*( X00*np.cos(ooA10(t)) +
Y00*np.sin(ooA10(t)) + X11*np.cos(iiA10(t)) + Y11*np.sin(iiA10(t)) )

# Constantes de Runge-Kutta
# Nota: K hace referencia a reales y L a Complejos
for i in range(0,n-1):

    T[i+1] = T[i] + h

    if (bell == 1):
    if ( T[i+1] <= tf1 ):
        w = E01 - E00

    elif( T[i+1] > tf1 and T[i+1] <= tf2 ):
        w = E11 - E01

    elif ( T[i+1] > tf2 and T[i+1] <= tf ):
        w = 0

    elif ( T[i+1] > tf ):
        break

    elif(bell == 2):
    T[i+1] = T[i] + h

    if ( T[i+1] <= tf1 ):
        w = E11 - E01

    elif( T[i+1] > tf1 and T[i+1] <= tf2 ):
        w = E11 - E10

    elif ( T[i+1] > tf2 and T[i+1] <= tf ):
        w = 0

    elif ( T[i+1] > tf ):
        break

```

```

elif(bell == 3):
T[i+1] = T[i] + h

if ( T[i+1] <= tf1 ):
    w = E10 - E00

elif( T[i+1] > tf1 and T[i+1] <= tf2 ):
    w = E11 - E10

elif ( T[i+1] > tf2 and T[i+1] <= tf ):
    w = 0

elif ( T[i+1] > tf ):
    break

elif(bell == 4):
if ( T[i+1] <= tf1 ):
    w = E11 - E01

elif( T[i+1] > tf1 and T[i+1] <= tf2 ):
    w = E11 - E10

elif ( T[i+1] > tf2 and T[i+1] <= tf ):
    w = 0

elif ( T[i+1] > tf ):
    break

# Constantes (referenciadas por la letra a):
K00a = h*dX00(T[i], vX01[i], vX10[i], vY01[i], vY10[i])
L00a = h*dY00(T[i], vX01[i], vX10[i], vY01[i], vY10[i])
Kiia = h*dX11(T[i], vX01[i], vX10[i], vY01[i], vY10[i])
Liia = h*dY11(T[i], vX01[i], vX10[i], vY01[i], vY10[i])

K01a = h*dX01(T[i], vX00[i], vX11[i], vY00[i], vY11[i])
L01a = h*dY01(T[i], vX00[i], vX11[i], vY00[i], vY11[i])
K10a = h*dX10(T[i], vX00[i], vX11[i], vY00[i], vY11[i])
L10a = h*dY10(T[i], vX00[i], vX11[i], vY00[i], vY11[i])

# Constantes (referenciadas por la letra b):
K00b = h*dX00(T[i] + h/2, vX01[i] + K01a/2, vX10[i] + K10a/2,
vY01[i] + L01a/2, vY10[i] + L10a/2)
L00b = h*dY00(T[i] + h/2, vX01[i] + K01a/2, vX10[i] + K10a/2,
vY01[i] + L01a/2, vY10[i] + L10a/2)
K11b = h*dX11(T[i] + h/2, vX01[i] + K01a/2, vX10[i] + K10a/2,
vY01[i] + L01a/2, vY10[i] + L10a/2)
L11b = h*dY11(T[i] + h/2, vX01[i] + K01a/2, vX10[i] + K10a/2,
vY01[i] + L01a/2, vY10[i] + L10a/2)

K01b = h*dX01(T[i] + h/2, vX00[i] + K00a/2, vX11[i] + Kiia/2,
vY00[i] + L00a/2, vY11[i] + Liia/2)
L01b = h*dY01(T[i] + h/2, vX00[i] + K00a/2, vX11[i] + Kiia/2,
vY00[i] + L00a/2, vY11[i] + Liia/2)
K10b = h*dX10(T[i] + h/2, vX00[i] + K00a/2, vX11[i] + Kiia/2,
vY00[i] + L00a/2, vY11[i] + Liia/2)

```

```

    L10b = h*dY10(T[i] + h/2, vX00[i] + K00a/2, vX11[i] + Kiia/2,
vY00[i] + L00a/2, vY11[i] + Liia/2)

# Constantes (referenciadas por la letra c):
    K00c = h*dX00(T[i] + h/2, vX01[i] + K01b/2, vX10[i] + K10b/2,
vY01[i] + L01b/2, vY10[i] + L10b/2)
    L00c = h*dY00(T[i] + h/2, vX01[i] + K01b/2, vX10[i] + K10b/2,
vY01[i] + L01b/2, vY10[i] + L10b/2)
    K11c = h*dX11(T[i] + h/2, vX01[i] + K01b/2, vX10[i] + K10b/2,
vY01[i] + L01b/2, vY10[i] + L10b/2)
    L11c = h*dY11(T[i] + h/2, vX01[i] + K01b/2, vX10[i] + K10b/2,
vY01[i] + L01b/2, vY10[i] + L10b/2)

    K01c = h*dX01(T[i] + h/2, vX00[i] + K00b/2, vX11[i] + K11b/2,
vY00[i] + L00b/2, vY11[i] + L11b/2)
    L01c = h*dY01(T[i] + h/2, vX00[i] + K00b/2, vX11[i] + K11b/2,
vY00[i] + L00b/2, vY11[i] + L11b/2)
    K10c = h*dX10(T[i] + h/2, vX00[i] + K00b/2, vX11[i] + K11b/2,
vY00[i] + L00b/2, vY11[i] + L11b/2)
    L10c = h*dY10(T[i] + h/2, vX00[i] + K00b/2, vX11[i] + K11b/2,
vY00[i] + L00b/2, vY11[i] + L11b/2)

# Constantes (referenciadas por la letra d):
    K00d = h*dX00(T[i] + h, vX01[i] + K01c, vX10[i] + K10c, vY01[i] +
L01c, vY10[i] + L10c)
    L00d = h*dY00(T[i] + h, vX01[i] + K01c, vX10[i] + K10c, vY01[i] +
L01c, vY10[i] + L10c)
    K11d = h*dX11(T[i] + h, vX01[i] + K01c, vX10[i] + K10c, vY01[i] +
L01c, vY10[i] + L10c)
    L11d = h*dY11(T[i] + h, vX01[i] + K01c, vX10[i] + K10c, vY01[i] +
L01c, vY10[i] + L10c)

    K01d = h*dX01(T[i] + h, vX00[i] + K00c, vX11[i] + K11c, vY00[i] +
L00c, vY11[i] + L11c)
    L01d = h*dY01(T[i] + h, vX00[i] + K00c, vX11[i] + K11c, vY00[i] +
L00c, vY11[i] + L11c)
    K10d = h*dX10(T[i] + h, vX00[i] + K00c, vX11[i] + K11c, vY00[i] +
L00c, vY11[i] + L11c)
    L10d = h*dY10(T[i] + h, vX00[i] + K00c, vX11[i] + K11c, vY00[i] +
L00c, vY11[i] + L11c)

# Soluciones para el esquema de interaccion por Runge-Kutta
    vX00[i+1] = vX00[i] + (1/6)*(K00a + 2*K00b + 2*K00c + K00d)
    vY00[i+1] = vY00[i] + (1/6)*(L00a + 2*L00b + 2*L00c + L00d)

    vX01[i+1] = vX01[i] + (1/6)*(K01a + 2*K01b + 2*K01c + K01d)
    vY01[i+1] = vY01[i] + (1/6)*(L01a + 2*L01b + 2*L01c + L01d)

    vX10[i+1] = vX10[i] + (1/6)*(K10a + 2*K10b + 2*K10c + K10d)
    vY10[i+1] = vY10[i] + (1/6)*(L10a + 2*L10b + 2*L10c + L10d)

    vX11[i+1] = vX11[i] + (1/6)*(Kiia + 2*K11b + 2*K11c + K11d)
    vY11[i+1] = vY11[i] + (1/6)*(Liia + 2*L11b + 2*L11c + L11d)

# Soluciones para la funcion de onda
    cX00[i+1] = vX00[i+1]*cos(E00*T[i+1]) - vY00[i+1]*sin(E00*T[i+1])
    cY00[i+1] = vX00[i+1]*sin(E00*T[i+1]) + vY00[i+1]*cos(E00*T[i+1])

```

```

cX01[i+1] = vX01[i+1]*cos(E01*T[i+1]) - vY01[i+1]*sin(E01*T[i+1])
cY01[i+1] = vX01[i+1]*sin(E01*T[i+1]) + vY01[i+1]*cos(E01*T[i+1])

cX10[i+1] = vX10[i+1]*cos(E10*T[i+1]) - vY10[i+1]*sin(E10*T[i+1])
cY10[i+1] = vX10[i+1]*sin(E10*T[i+1]) + vY10[i+1]*cos(E10*T[i+1])

cX11[i+1] = vX11[i+1]*cos(E11*T[i+1]) - vY11[i+1]*sin(E11*T[i+1])
cY11[i+1] = vX11[i+1]*sin(E11*T[i+1]) + vY11[i+1]*cos(E11*T[i+1])

# Determinacion de probabilidades

D00[i] = (cX00[i])**2 + (cY00[i])**2
D01[i] = (cX01[i])**2 + (cY01[i])**2
D10[i] = (cX10[i])**2 + (cY10[i])**2
D11[i] = (cX11[i])**2 + (cY11[i])**2

# Comprobacion de normalidad
N[i] = D00[i] + D01[i] + D10[i] + D11[i]

D00[-1] = (cX00[-1])**2 + (cY00[-1])**2
D01[-1] = (cX01[-1])**2 + (cY01[-1])**2
D10[-1] = (cX10[-1])**2 + (cY10[-1])**2
D11[-1] = (cX11[-1])**2 + (cY11[-1])**2

N[-1] = D00[-1] + D01[-1] + D10[-1] + D11[-1]

# Graficar la comprobacion del sistema.

sstyle = "seaborn-poster"
plt.style.use(sstyle)
plt.rc('text', usetex=True)
plt.rc('font', family='serif')
plt.figure(figsize = (20,10))

file_name = "X00_" + "%.2f"%vX00[0] + " Y00_" + "%.2f"%vY00[0] + \
" X01_" + "%.2f"%vX01[0] + " Y01_" + "%.2f"%vY01[0] + \
" X10_" + "%.2f"%vX10[0] + " Y10_" + "%.2f"%vY10[0] + \
" X11_" + "%.2f"%vX11[0] + " Y11_" + "%.2f"%vY11[0] + \
" U_" + "%.2f"%U + " w_" + "%.2f"%w + \
" w1_" + "%.2f"%w1 + " w2_" + "%.2f"%w2 + \
" J_" + "%.2f"%J + " phi_" + "%.2f"%phi + \
" h_" + "%.4f"%h

points = int(0.06*n)
y_ticks = np.arange(0, 1.1, step=0.1)

plt.plot(T, N, label = "Probabilidad Total")
plt.plot(T, D00, label = "$|C_{00}(t)|^2$", marker="o", markevery=points)
plt.plot(T, D01, label = "$|C_{01}(t)|^2$", marker="v", markevery=points)
plt.plot(T, D10, label = "$|C_{10}(t)|^2$", marker="d", markevery=points)
plt.plot(T, D11, label = "$|C_{11}(t)|^2$", marker="p", markevery=points)
plt.vlines([tf1, tf2], 0, 1, colors='k')

plt.yticks(y_ticks)

if (bell == 1):
FO-DGA-CPAP-0017

```

```

plt.title("Estado de Bell $ | \\beta_{00} \\rangle $")

elif (bell == 2):
plt.title("Estado de Bell $ | \\beta_{01} \\rangle $")

elif (bell == 3):
plt.title("Estado de Bell $ | \\beta_{10} \\rangle $")

elif (bell == 4):
plt.title("Estado de Bell $ | \\beta_{11} \\rangle $")

plt.xlabel("Tiempo (s)")
plt.ylabel("Probabilidad")

ax = plt.subplot(111)
box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width*0.8, box.height])
plt.legend(loc=1, bbox_to_anchor=(1.4, 0.7), frameon=False)
plt.grid()

#plt.savefig( file_name + ".png", format = "png")
plt.show()

# COEFICIENTES DE LA FUNCION DE ONDA

XMAX = np.array( [ max(cX00), max(cX01), max(cX10), max(cX11) ] )
YMAX = np.array( [ max(cY00), max(cY01), max(cY10), max(cY11) ] )
TMAX = max( np.concatenate( (XMAX,YMAX), axis = None ) ) + 0.1

XMIN = np.array( [ min(cX00), min(cX01), min(cX10), min(cX11) ] )
YMIN = np.array( [ min(cY00), min(cY01), min(cY10), min(cY11) ] )
TMIN = min( np.concatenate( (XMIN,YMIN), axis = None ) ) - 0.1

if(COMP == 'N' or COMP == 'n' or COMP == 'No' or COMP == 'NO' or COMP ==
'no' or COMP == 'Nope' or COMP == 'nope'):
pass

elif (COMP == 'Y' or COMP == 'y' or COMP == 'Yes' or COMP == 'YES' or
COMP == 'yes'):

# -----
# PRIMER COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX00, label = "$X_{00}$ - Real ")
plt.plot(T, cY00, label = "$Y_{00}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{00}(t)$")
plt.xlabel("Time (s)")
plt.ylabel("Probability")
plt.legend(loc=0)
plt.grid()
plt.show()

```

```

# -----
# SEGUNDO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX01, label = "$X_{01}$ - Real ")
plt.plot(T, cY01, label = "$Y_{01}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{01}(t)$")
plt.xlabel("Time (s)")
plt.ylabel("Probability")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# TERCER COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX10, label = "$X_{10}$ - Real ")
plt.plot(T, cY10, label = "$Y_{10}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{10}(t)$")
plt.xlabel("Time (s)")
plt.ylabel("Probability")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# CUARTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX11, label = "$X_{11}$ - Real ")
plt.plot(T, cY11, label = "$Y_{11}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{11}(t)$")
plt.xlabel("Time (s)")
plt.ylabel("Probability")
plt.legend(loc=0)
plt.grid()
plt.show()

```

Anexo 6. Código en PYTHON para Teleportación Cuántica

```
"""
PROGRAMA QUE RESUELVE NUMERICAMENTE LAS FUNCIONES  $D_{000}(t)$ , ...,  $D_{111}(t)$ 
QUE SE OBTIENEN AL TRATAR LA ECUACION DEL ESQUEMA DE INTERACCION PARA TRES
QUBITS DE CARBONO C13 EN UN ENLACE DE DIAMANTE BAJO LA INFLUENCIA DE UN
CAMPO MAGNETICO Y LA SIMULACION DE ENTRELAZAMIENTO CUNATICO PARA GENERAR
TELETRANSPORTANCION CUANTICA.

15/11/2018 - 20/11/2018
QTeleportationP.py
"""

import numpy as np
from numpy import sin, cos, sqrt, pi
import matplotlib.pyplot as plt
from matplotlib.pyplot import savefig
import matplotlib.path_effects as path_effects

# CONSTANTES DEL SISTEMA
w0 = 1/pi # FRECUENCIA DEL TIEMPO TAO

U = 0.1/w0 # FRECUENCIA DE RABI
w1 = 100/w0 # FRECUENCIA DE LARMOR DEL PRIMER QUBIT
w2 = 250/w0 # FRECUENCIA DE LARMOR DEL SEGUNDO QUBIT
w3 = 500/w0 # FRECUENCIA DE LARMOR DEL TERCER QUBIT
J = 5.0/w0 # FACTOR DE INTERACCION ENTRE PRIMEROS VECINOS (FRECUENCIA)
Jp = 0.8/w0 # FACTOR DE INTERACCION ENTRE SEGUNDOS VECINOS (FRECUENCIA)

phi = 0.0 # FASE DEL CAMPO MAGNETICO

pp = pi/U #PI PULSO
hpp = pi/(2*U) # PI MEDIO PULSO

# DELIMITACIONES DEL METODO NUMERICOC.
ti = 0 # TIEMPO INICIAL: siempre cero
tf1 = hpp # TIEMPOS FINALES DEFINIDOS POR PI-PULSOS
tf2 = tf1 + hpp
tf3 = tf2 + pp
tf4 = tf3 + pp
tf5 = tf4 + hpp
tf6 = tf5 + hpp
tf7 = tf6 + hpp
tf8 = tf7 + hpp
tf = tf8 + hpp # TIEMPO FINAL

h = 0.001 # TAMANHO DE PASO PARA POR CADA CALCULO
n = int((tf-ti)/h) # NUMERO DE ITERACIONES TOTALES PARA EL CODIGO
w = 0

# ENERGIA DEL SISTEMA POR UNIDAD h-BARRA.

E_hbar = lambda a3, a2, a1: (-1/2)* ( (w1)*((-1)**a1) + (w2)*((-1)**a2) +
(w3)*((-1)**a3) + 0.5*(J)*( ( (-1)**(a1+a2) ) + ( (-1)**(a2+a3) ) ) +
0.5*(Jp)*( (-1)**(a1+a3) ) )

FO-DGA-CPAP-0017
```

```

E = np.zeros(8)
E[0] = E_hbar(0,0,0)      #ENERGIA DEL ESTADO 000
E[1] = E_hbar(0,0,1)      #ENERGIA DEL ESTADO 001
E[2] = E_hbar(0,1,0)      #ENERGIA DEL ESTADO 010
E[3] = E_hbar(0,1,1)      #ENERGIA DEL ESTADO 011
E[4] = E_hbar(1,0,0)      #ENERGIA DEL ESTADO 100
E[5] = E_hbar(1,0,1)      #ENERGIA DEL ESTADO 101
E[6] = E_hbar(1,1,0)      #ENERGIA DEL ESTADO 110
E[7] = E_hbar(1,1,1)      #ENERGIA DEL ESTADO 111

Wij = lambda i, j: (E[i-1] - E[j-1])
theta = lambda t: w*t + phi      # Angulo del campo magnetico
B = lambda t, i, j: Wij(i,j)*t + theta(t)

# ARREGLOS PARA SOLUCION NUMERICA

vX1, vY1 = np.zeros(n), np.zeros(n)
vX2, vY2 = np.zeros(n), np.zeros(n)
vX3, vY3 = np.zeros(n), np.zeros(n)
vX4, vY4 = np.zeros(n), np.zeros(n)
vX5, vY5 = np.zeros(n), np.zeros(n)
vX6, vY6 = np.zeros(n), np.zeros(n)
vX7, vY7 = np.zeros(n), np.zeros(n)
vX8, vY8 = np.zeros(n), np.zeros(n)

cX1, cY1 = np.zeros(n), np.zeros(n)
cX2, cY2 = np.zeros(n), np.zeros(n)
cX3, cY3 = np.zeros(n), np.zeros(n)
cX4, cY4 = np.zeros(n), np.zeros(n)
cX5, cY5 = np.zeros(n), np.zeros(n)
cX6, cY6 = np.zeros(n), np.zeros(n)
cX7, cY7 = np.zeros(n), np.zeros(n)
cX8, cY8 = np.zeros(n), np.zeros(n)

D1 = np.zeros(n)
D2 = np.zeros(n)
D3 = np.zeros(n)
D4 = np.zeros(n)
D5 = np.zeros(n)
D6 = np.zeros(n)
D7 = np.zeros(n)
D8 = np.zeros(n)

N = np.zeros(n)
T = np.zeros(n)

# CONDICIONES INICIALES DEL SISTEMA DE ECUACIONES DEL MARCO DE
INTERACCION
vX1[0], vY1[0] = sqrt(1/2), 0
vX2[0], vY2[0] = 0, 0
vX3[0], vY3[0] = 0, 0
vX4[0], vY4[0] = 0, 0
vX5[0], vY5[0] = sqrt(1/2), 0
vX6[0], vY6[0] = 0, 0
vX7[0], vY7[0] = 0, 0
FO-DGA-CPAP-0017

```

$vX8[0], vY8[0] = 0, 0$

CONDICIONES INICIALES DEL SISTEMA DE ECUACIONES DE LA FUNCION DE ONDA

$cX1[0], cY1[0] = vX1[0], vY1[0]$
 $cX2[0], cY2[0] = vX2[0], vY2[0]$
 $cX3[0], cY3[0] = vX3[0], vY3[0]$
 $cX4[0], cY4[0] = vX4[0], vY4[0]$
 $cX5[0], cY5[0] = vX5[0], vY5[0]$
 $cX6[0], cY6[0] = vX6[0], vY6[0]$
 $cX7[0], cY7[0] = vX7[0], vY7[0]$
 $cX8[0], cY8[0] = vX8[0], vY8[0]$

SISTEMA DE ECUACIONES DIFERENCIALES DEL MARCO DE INTERACCION

$dx1 = \text{lambda } t, X2, Y2, X3, Y3, X5, Y5: (U/2)*(-X2*\sin(B(t,1,2)) + -Y2*\cos(B(t,1,2)) + -X3*\sin(B(t,1,3)) + -Y3*\cos(B(t,1,3)) + -X5*\sin(B(t,1,5)) + -Y5*\cos(B(t,1,5)))$

$dy1 = \text{lambda } t, X2, Y2, X3, Y3, X5, Y5: (U/2)*(X2*\cos(B(t,1,2)) + -Y2*\sin(B(t,1,2)) + X3*\cos(B(t,1,3)) + -Y3*\sin(B(t,1,3)) + X5*\cos(B(t,1,5)) + -Y5*\sin(B(t,1,5)))$

$dx2 = \text{lambda } t, X1, Y1, X4, Y4, X6, Y6: (U/2)*(X1*\sin(B(t,1,2)) + -Y1*\cos(B(t,1,2)) + -X4*\sin(B(t,2,4)) + -Y4*\cos(B(t,2,4)) + -X6*\sin(B(t,2,6)) + -Y6*\cos(B(t,2,6)))$

$dy2 = \text{lambda } t, X1, Y1, X4, Y4, X6, Y6: (U/2)*(X1*\cos(B(t,1,2)) + Y1*\sin(B(t,1,2)) + X4*\cos(B(t,2,4)) + -Y4*\sin(B(t,2,4)) + X6*\cos(B(t,2,6)) + -Y6*\sin(B(t,2,6)))$

$dx3 = \text{lambda } t, X1, Y1, X4, Y4, X7, Y7: (U/2)*(X1*\sin(B(t,1,3)) + -Y1*\cos(B(t,1,3)) + -X4*\sin(B(t,3,4)) + -Y4*\cos(B(t,3,4)) + -X7*\sin(B(t,3,7)) + -Y7*\cos(B(t,3,7)))$

$dy3 = \text{lambda } t, X1, Y1, X4, Y4, X7, Y7: (U/2)*(X1*\cos(B(t,1,3)) + Y1*\sin(B(t,1,3)) + X4*\cos(B(t,3,4)) + -Y4*\sin(B(t,3,4)) + X7*\cos(B(t,3,7)) + -Y7*\sin(B(t,3,7)))$

$dx4 = \text{lambda } t, X2, Y2, X3, Y3, X8, Y8: (U/2)*(X2*\sin(B(t,2,4)) + -Y2*\cos(B(t,2,4)) + X3*\sin(B(t,3,4)) + -Y3*\cos(B(t,3,4)) + -X8*\sin(B(t,4,8)) + -Y8*\cos(B(t,4,8)))$

$dy4 = \text{lambda } t, X2, Y2, X3, Y3, X8, Y8: (U/2)*(X2*\cos(B(t,2,4)) + Y2*\sin(B(t,2,4)) + X3*\cos(B(t,3,4)) + Y3*\sin(B(t,3,4)) + X8*\cos(B(t,4,8)) + -Y8*\sin(B(t,4,8)))$

$dx5 = \text{lambda } t, X1, Y1, X6, Y6, X7, Y7: (U/2)*(X1*\sin(B(t,1,5)) + -Y1*\cos(B(t,1,5)) + -X6*\sin(B(t,5,6)) + -Y6*\cos(B(t,5,6)) + -X7*\sin(B(t,5,7)) + -Y7*\cos(B(t,5,7)))$

$dy5 = \text{lambda } t, X1, Y1, X6, Y6, X7, Y7: (U/2)*(X1*\cos(B(t,1,5)) + Y1*\sin(B(t,1,5)) + X6*\cos(B(t,5,6)) + -Y6*\sin(B(t,5,6)) + X7*\cos(B(t,5,7)) + -Y7*\sin(B(t,5,7)))$

$dx6 = \text{lambda } t, X2, Y2, X5, Y5, X8, Y8: (U/2)*(X2*\sin(B(t,2,6)) + -Y2*\cos(B(t,2,6)) + X5*\sin(B(t,5,6)) + -Y5*\cos(B(t,5,6)) + -X8*\sin(B(t,6,8)) + -Y8*\cos(B(t,6,8)))$

$dy6 = \text{lambda } t, X2, Y2, X5, Y5, X8, Y8: (U/2)*(X2*\cos(B(t,2,6)) + Y2*\sin(B(t,2,6)) + X5*\cos(B(t,5,6)) + Y5*\sin(B(t,5,6)) + X8*\cos(B(t,6,8)) + -Y8*\sin(B(t,6,8)))$

```

dX7 = lambda t, X3, Y3, X5, Y5, X8, Y8: (U/2)*( X3*sin(B(t,3,7)) + -
Y3*cos(B(t,3,7)) + X5*sin(B(t,5,7)) + -Y5*cos(B(t,5,7)) + -
X8*sin(B(t,7,8)) + -Y8*cos(B(t,7,8)) )
dY7 = lambda t, X3, Y3, X5, Y5, X8, Y8: (U/2)*( X3*cos(B(t,3,7)) +
Y3*sin(B(t,3,7)) + X5*cos(B(t,5,7)) + Y5*sin(B(t,5,7)) +
X8*cos(B(t,7,8)) + -Y8*sin(B(t,7,8)) )

dX8 = lambda t, X4, Y4, X6, Y6, X7, Y7: (U/2)*( X4*sin(B(t,4,8)) + -
Y4*cos(B(t,4,8)) + X6*sin(B(t,6,8)) + -Y6*cos(B(t,6,8))
+ X7*sin(B(t,7,8)) + -Y7*cos(B(t,7,8)) )
dY8 = lambda t, X4, Y4, X6, Y6, X7, Y7: (U/2)*( X4*cos(B(t,4,8)) +
Y4*sin(B(t,4,8)) + X6*cos(B(t,6,8)) + Y6*sin(B(t,6,8)) +
X7*cos(B(t,7,8)) + Y7*sin(B(t,7,8)) )

# COEFICIENTES K Y L DEL METODO RUNGE-KUTTA CUARTO ORDEN
# Noca: K hace referencia a reales y L a Complejos

K = np.zeros((5,9))
L = np.zeros((5,9))

for i in range(0,n-1):

    T[i+1] = T[i] + h

    if ( T[i+1] <= tf1 ):
        w = Wij(3,1)

    elif ( T[i+1] > tf1 and T[i+1] <= tf2 ):
        w = Wij(7,5)

    elif ( T[i+1] > tf2 and T[i+1] <= tf3 ):
        w = Wij(4,3)

    elif ( T[i+1] > tf3 and T[i+1] <= tf4 ):
        w = Wij(6,5)

    elif ( T[i+1] > tf4 and T[i+1] <= tf5 ):
        w = Wij(5,1)

    elif ( T[i+1] > tf5 and T[i+1] <= tf6 ):
        w = Wij(6,2)

    elif ( T[i+1] > tf6 and T[i+1] <= tf7 ):
        w = Wij(8,4)

    elif ( T[i+1] > tf7 and T[i+1] <= tf8 ):
        w = Wij(7,3)

    elif ( T[i+1] > tf8 and T[i+1] <= tf ):
        w = 0

    elif ( T[i+1] > tf ):
        break # El codigo continua afuera del 'for i in range(0,n-1)'

    K[1,1] = h*dX1( T[i], vX2[i], vY2[i], vX3[i], vY3[i], vX5[i], vY5[i]
)

```

$K[1,2] = h \cdot dx2(T[i], vx1[i], vy1[i], vx4[i], vy4[i], vx6[i], vy6[i]$
 $)$
 $K[1,3] = h \cdot dx3(T[i], vx1[i], vy1[i], vx4[i], vy4[i], vx7[i], vy7[i]$
 $)$
 $K[1,4] = h \cdot dx4(T[i], vx2[i], vy2[i], vx3[i], vy3[i], vx8[i], vy8[i]$
 $)$
 $K[1,5] = h \cdot dx5(T[i], vx1[i], vy1[i], vx6[i], vy6[i], vx7[i], vy7[i]$
 $)$
 $K[1,6] = h \cdot dx6(T[i], vx2[i], vy2[i], vx5[i], vy5[i], vx8[i], vy8[i]$
 $)$
 $K[1,7] = h \cdot dx7(T[i], vx3[i], vy3[i], vx5[i], vy5[i], vx8[i], vy8[i]$
 $)$
 $K[1,8] = h \cdot dx8(T[i], vx4[i], vy4[i], vx6[i], vy6[i], vx7[i], vy7[i]$
 $)$

$L[1,1] = h \cdot dy1(T[i], vx2[i], vy2[i], vx3[i], vy3[i], vx5[i], vy5[i]$
 $)$
 $L[1,2] = h \cdot dy2(T[i], vx1[i], vy1[i], vx4[i], vy4[i], vx6[i], vy6[i]$
 $)$
 $L[1,3] = h \cdot dy3(T[i], vx1[i], vy1[i], vx4[i], vy4[i], vx7[i], vy7[i]$
 $)$
 $L[1,4] = h \cdot dy4(T[i], vx2[i], vy2[i], vx3[i], vy3[i], vx8[i], vy8[i]$
 $)$
 $L[1,5] = h \cdot dy5(T[i], vx1[i], vy1[i], vx6[i], vy6[i], vx7[i], vy7[i]$
 $)$
 $L[1,6] = h \cdot dy6(T[i], vx2[i], vy2[i], vx5[i], vy5[i], vx8[i], vy8[i]$
 $)$
 $L[1,7] = h \cdot dy7(T[i], vx3[i], vy3[i], vx5[i], vy5[i], vx8[i], vy8[i]$
 $)$
 $L[1,8] = h \cdot dy8(T[i], vx4[i], vy4[i], vx6[i], vy6[i], vx7[i], vy7[i]$
 $)$

$K[2,1] = h \cdot dx1(T[i] + h/2, vx2[i] + K[1,2]/2, vy2[i] + L[1,2]/2,$
 $vx3[i] + K[1,3]/2, vy3[i] + L[1,3]/2, vx5[i] + K[1,5]/2, vy5[i] +$
 $L[1,5]/2)$

$K[2,2] = h \cdot dx2(T[i] + h/2, vx1[i] + K[1,1]/2, vy1[i] + L[1,1]/2,$
 $vx4[i] + K[1,4]/2, vy4[i] + L[1,4]/2, vx6[i] + K[1,6]/2, vy6[i] +$
 $L[1,6]/2)$

$K[2,3] = h \cdot dx3(T[i] + h/2, vx1[i] + K[1,1]/2, vy1[i] + L[1,1]/2,$
 $vx4[i] + K[1,4]/2, vy4[i] + L[1,4]/2, vx7[i] + K[1,7]/2, vy7[i] +$
 $L[1,7]/2)$

$K[2,4] = h \cdot dx4(T[i] + h/2, vx2[i] + K[1,2]/2, vy2[i] + L[1,2]/2,$
 $vx3[i] + K[1,3]/2, vy3[i] + L[1,3]/2, vx8[i] + K[1,8]/2, vy8[i] +$
 $L[1,8]/2)$

$K[2,5] = h \cdot dx5(T[i] + h/2, vx1[i] + K[1,1]/2, vy1[i] + L[1,1]/2,$
 $vx6[i] + K[1,6]/2, vy6[i] + L[1,6]/2, vx7[i] + K[1,7]/2, vy7[i] +$
 $L[1,7]/2)$

$K[2,6] = h \cdot dx6(T[i] + h/2, vx2[i] + K[1,2]/2, vy2[i] + L[1,2]/2,$
 $vx5[i] + K[1,5]/2, vy5[i] + L[1,5]/2, vx8[i] + K[1,8]/2, vy8[i] +$
 $L[1,8]/2)$

$K[2,7] = h \cdot dx7(T[i] + h/2, vx3[i] + K[1,3]/2, vy3[i] + L[1,3]/2,$
 $vx5[i] + K[1,5]/2, vy5[i] + L[1,5]/2, vx8[i] + K[1,8]/2, vy8[i] +$
 $L[1,8]/2)$

$$K[2,8] = h*dX8(T[i] + h/2, vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX6[i] + K[1,6]/2, vY6[i] + L[1,6]/2, vX7[i] + K[1,7]/2, vY7[i] + L[1,7]/2)$$

$$L[2,1] = h*dY1(T[i] + h/2, vX2[i] + K[1,2]/2, vY2[i] + L[1,2]/2, vX3[i] + K[1,3]/2, vY3[i] + L[1,3]/2, vX5[i] + K[1,5]/2, vY5[i] + L[1,5]/2)$$

$$L[2,2] = h*dY2(T[i] + h/2, vX1[i] + K[1,1]/2, vY1[i] + L[1,1]/2, vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX6[i] + K[1,6]/2, vY6[i] + L[1,6]/2)$$

$$L[2,3] = h*dY3(T[i] + h/2, vX1[i] + K[1,1]/2, vY1[i] + L[1,1]/2, vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX7[i] + K[1,7]/2, vY7[i] + L[1,7]/2)$$

$$L[2,4] = h*dY4(T[i] + h/2, vX2[i] + K[1,2]/2, vY2[i] + L[1,2]/2, vX3[i] + K[1,3]/2, vY3[i] + L[1,3]/2, vX8[i] + K[1,8]/2, vY8[i] + L[1,8]/2)$$

$$L[2,5] = h*dY5(T[i] + h/2, vX1[i] + K[1,1]/2, vY1[i] + L[1,1]/2, vX6[i] + K[1,6]/2, vY6[i] + L[1,6]/2, vX7[i] + K[1,7]/2, vY7[i] + L[1,7]/2)$$

$$L[2,6] = h*dY6(T[i] + h/2, vX2[i] + K[1,2]/2, vY2[i] + L[1,2]/2, vX5[i] + K[1,5]/2, vY5[i] + L[1,5]/2, vX8[i] + K[1,8]/2, vY8[i] + L[1,8]/2)$$

$$L[2,7] = h*dY7(T[i] + h/2, vX3[i] + K[1,3]/2, vY3[i] + L[1,3]/2, vX5[i] + K[1,5]/2, vY5[i] + L[1,5]/2, vX8[i] + K[1,8]/2, vY8[i] + L[1,8]/2)$$

$$L[2,8] = h*dY8(T[i] + h/2, vX4[i] + K[1,4]/2, vY4[i] + L[1,4]/2, vX6[i] + K[1,6]/2, vY6[i] + L[1,6]/2, vX7[i] + K[1,7]/2, vY7[i] + L[1,7]/2)$$

$$K[3,1] = h*dX1(T[i] + h/2, vX2[i] + K[2,2]/2, vY2[i] + L[2,2]/2, vX3[i] + K[2,3]/2, vY3[i] + L[2,3]/2, vX5[i] + K[2,5]/2, vY5[i] + L[2,5]/2)$$

$$K[3,2] = h*dX2(T[i] + h/2, vX1[i] + K[2,1]/2, vY1[i] + L[2,1]/2, vX4[i] + K[2,4]/2, vY4[i] + L[2,4]/2, vX6[i] + K[2,6]/2, vY6[i] + L[2,6]/2)$$

$$K[3,3] = h*dX3(T[i] + h/2, vX1[i] + K[2,1]/2, vY1[i] + L[2,1]/2, vX4[i] + K[2,4]/2, vY4[i] + L[2,4]/2, vX7[i] + K[2,7]/2, vY7[i] + L[2,7]/2)$$

$$K[3,4] = h*dX4(T[i] + h/2, vX2[i] + K[2,2]/2, vY2[i] + L[2,2]/2, vX3[i] + K[2,3]/2, vY3[i] + L[2,3]/2, vX8[i] + K[2,8]/2, vY8[i] + L[2,8]/2)$$

$$K[3,5] = h*dX5(T[i] + h/2, vX1[i] + K[2,1]/2, vY1[i] + L[2,1]/2, vX6[i] + K[2,6]/2, vY6[i] + L[2,6]/2, vX7[i] + K[2,7]/2, vY7[i] + L[2,7]/2)$$

$$K[3,6] = h*dX6(T[i] + h/2, vX2[i] + K[2,2]/2, vY2[i] + L[2,2]/2, vX5[i] + K[2,5]/2, vY5[i] + L[2,5]/2, vX8[i] + K[2,8]/2, vY8[i] + L[2,8]/2)$$

$$K[3,7] = h*dX7(T[i] + h/2, vX3[i] + K[2,3]/2, vY3[i] + L[2,3]/2, vX5[i] + K[2,5]/2, vY5[i] + L[2,5]/2, vX8[i] + K[2,8]/2, vY8[i] + L[2,8]/2)$$

$$K[3,8] = h*dX8(T[i] + h/2, vX4[i] + K[2,4]/2, vY4[i] + L[2,4]/2, vX6[i] + K[2,6]/2, vY6[i] + L[2,6]/2, vX7[i] + K[2,7]/2, vY7[i] + L[2,7]/2)$$

$L[3,1] = h*dY1(T[i] + h/2, vX2[i] + K[2,2]/2, vY2[i] + L[2,2]/2, vX3[i] + K[2,3]/2, vY3[i] + L[2,3]/2, vX5[i] + K[2,5]/2, vY5[i] + L[2,5]/2)$
 $L[3,2] = h*dY2(T[i] + h/2, vX1[i] + K[2,1]/2, vY1[i] + L[2,1]/2, vX4[i] + K[2,4]/2, vY4[i] + L[2,4]/2, vX6[i] + K[2,6]/2, vY6[i] + L[2,6]/2)$
 $L[3,3] = h*dY3(T[i] + h/2, vX1[i] + K[2,1]/2, vY1[i] + L[2,1]/2, vX4[i] + K[2,4]/2, vY4[i] + L[2,4]/2, vX7[i] + K[2,7]/2, vY7[i] + L[2,7]/2)$
 $L[3,4] = h*dY4(T[i] + h/2, vX2[i] + K[2,2]/2, vY2[i] + L[2,2]/2, vX3[i] + K[2,3]/2, vY3[i] + L[2,3]/2, vX8[i] + K[2,8]/2, vY8[i] + L[2,8]/2)$
 $L[3,5] = h*dY5(T[i] + h/2, vX1[i] + K[2,1]/2, vY1[i] + L[2,1]/2, vX6[i] + K[2,6]/2, vY6[i] + L[2,6]/2, vX7[i] + K[2,7]/2, vY7[i] + L[2,7]/2)$
 $L[3,6] = h*dY6(T[i] + h/2, vX2[i] + K[2,2]/2, vY2[i] + L[2,2]/2, vX5[i] + K[2,5]/2, vY5[i] + L[2,5]/2, vX8[i] + K[2,8]/2, vY8[i] + L[2,8]/2)$
 $L[3,7] = h*dY7(T[i] + h/2, vX3[i] + K[2,3]/2, vY3[i] + L[2,3]/2, vX5[i] + K[2,5]/2, vY5[i] + L[2,5]/2, vX8[i] + K[2,8]/2, vY8[i] + L[2,8]/2)$
 $L[3,8] = h*dY8(T[i] + h/2, vX4[i] + K[2,4]/2, vY4[i] + L[2,4]/2, vX6[i] + K[2,6]/2, vY6[i] + L[2,6]/2, vX7[i] + K[2,7]/2, vY7[i] + L[2,7]/2)$

$K[4,1] = h*dX1(T[i] + h, vX2[i] + K[3,2], vY2[i] + L[3,2], vX3[i] + K[3,3], vY3[i] + L[3,3], vX5[i] + K[3,5], vY5[i] + L[3,5])$
 $K[4,2] = h*dX2(T[i] + h, vX1[i] + K[3,1], vY1[i] + L[3,1], vX4[i] + K[3,4], vY4[i] + L[3,4], vX6[i] + K[3,6], vY6[i] + L[3,6])$
 $K[4,3] = h*dX3(T[i] + h, vX1[i] + K[3,1], vY1[i] + L[3,1], vX4[i] + K[3,4], vY4[i] + L[3,4], vX7[i] + K[3,7], vY7[i] + L[3,7])$
 $K[4,4] = h*dX4(T[i] + h, vX2[i] + K[3,2], vY2[i] + L[3,2], vX3[i] + K[3,3], vY3[i] + L[3,3], vX8[i] + K[3,8], vY8[i] + L[3,8])$
 $K[4,5] = h*dX5(T[i] + h, vX1[i] + K[3,1], vY1[i] + L[3,1], vX6[i] + K[3,6], vY6[i] + L[3,6], vX7[i] + K[3,7], vY7[i] + L[3,7])$
 $K[4,6] = h*dX6(T[i] + h, vX2[i] + K[3,2], vY2[i] + L[3,2], vX5[i] + K[3,5], vY5[i] + L[3,5], vX8[i] + K[3,8], vY8[i] + L[3,8])$
 $K[4,7] = h*dX7(T[i] + h, vX3[i] + K[3,3], vY3[i] + L[3,3], vX5[i] + K[3,5], vY5[i] + L[3,5], vX8[i] + K[3,8], vY8[i] + L[3,8])$
 $K[4,8] = h*dX8(T[i] + h, vX4[i] + K[3,4], vY4[i] + L[3,4], vX6[i] + K[3,6], vY6[i] + L[3,6], vX7[i] + K[3,7], vY7[i] + L[3,7])$

$L[4,1] = h*dY1(T[i] + h, vX2[i] + K[3,2], vY2[i] + L[3,2], vX3[i] + K[3,3], vY3[i] + L[3,3], vX5[i] + K[3,5], vY5[i] + L[3,5])$
 $L[4,2] = h*dY2(T[i] + h, vX1[i] + K[3,1], vY1[i] + L[3,1], vX4[i] + K[3,4], vY4[i] + L[3,4], vX6[i] + K[3,6], vY6[i] + L[3,6])$
 $L[4,3] = h*dY3(T[i] + h, vX1[i] + K[3,1], vY1[i] + L[3,1], vX4[i] + K[3,4], vY4[i] + L[3,4], vX7[i] + K[3,7], vY7[i] + L[3,7])$
 $L[4,4] = h*dY4(T[i] + h, vX2[i] + K[3,2], vY2[i] + L[3,2], vX3[i] + K[3,3], vY3[i] + L[3,3], vX8[i] + K[3,8], vY8[i] + L[3,8])$
 $L[4,5] = h*dY5(T[i] + h, vX1[i] + K[3,1], vY1[i] + L[3,1], vX6[i] + K[3,6], vY6[i] + L[3,6], vX7[i] + K[3,7], vY7[i] + L[3,7])$
 $L[4,6] = h*dY6(T[i] + h, vX2[i] + K[3,2], vY2[i] + L[3,2], vX5[i] + K[3,5], vY5[i] + L[3,5], vX8[i] + K[3,8], vY8[i] + L[3,8])$

```

L[4,7] = h*dY7( T[i] + h, vX3[i] + K[3,3], vY3[i] + L[3,3], vX5[i] +
K[3,5], vY5[i] + L[3,5], vX8[i] + K[3,8], vY8[i] + L[3,8] )
L[4,8] = h*dY8( T[i] + h, vX4[i] + K[3,4], vY4[i] + L[3,4], vX6[i] +
K[3,6], vY6[i] + L[3,6], vX7[i] + K[3,7], vY7[i] + L[3,7] )

```

```

# SOLUCIONES A ECUACIONES DEL MARCO DE INTERACCION DADAS POR RUNGE-KUTTA
CUARTO ORDEN

```

```

vX1[i+1] = vX1[i] + (1/6)*( K[1,1] + 2*K[2,1] + 2*K[3,1] + K[4,1] )
vY1[i+1] = vY1[i] + (1/6)*( L[1,1] + 2*L[2,1] + 2*L[3,1] + L[4,1] )

vX2[i+1] = vX2[i] + (1/6)*( K[1,2] + 2*K[2,2] + 2*K[3,2] + K[4,2] )
vY2[i+1] = vY2[i] + (1/6)*( L[1,2] + 2*L[2,2] + 2*L[3,2] + L[4,2] )

vX3[i+1] = vX3[i] + (1/6)*( K[1,3] + 2*K[2,3] + 2*K[3,3] + K[4,3] )
vY3[i+1] = vY3[i] + (1/6)*( L[1,3] + 2*L[2,3] + 2*L[3,3] + L[4,3] )

vX4[i+1] = vX4[i] + (1/6)*( K[1,4] + 2*K[2,4] + 2*K[3,4] + K[4,4] )
vY4[i+1] = vY4[i] + (1/6)*( L[1,4] + 2*L[2,4] + 2*L[3,4] + L[4,4] )

vX5[i+1] = vX5[i] + (1/6)*( K[1,5] + 2*K[2,5] + 2*K[3,5] + K[4,5] )
vY5[i+1] = vY5[i] + (1/6)*( L[1,5] + 2*L[2,5] + 2*L[3,5] + L[4,5] )

vX6[i+1] = vX6[i] + (1/6)*( K[1,6] + 2*K[2,6] + 2*K[3,6] + K[4,6] )
vY6[i+1] = vY6[i] + (1/6)*( L[1,6] + 2*L[2,6] + 2*L[3,6] + L[4,6] )

vX7[i+1] = vX7[i] + (1/6)*( K[1,7] + 2*K[2,7] + 2*K[3,7] + K[4,7] )
vY7[i+1] = vY7[i] + (1/6)*( L[1,7] + 2*L[2,7] + 2*L[3,7] + L[4,7] )

vX8[i+1] = vX8[i] + (1/6)*( K[1,8] + 2*K[2,8] + 2*K[3,8] + K[4,8] )
vY8[i+1] = vY8[i] + (1/6)*( L[1,8] + 2*L[2,8] + 2*L[3,8] + L[4,8] )

```

```

# SOLUCIONES A ECUACIONES DE LA FUNCION DE ONDA DADAS POR RUNGE-KUTTA
CUARTO ORDEN

```

```

cX1[i+1] = vX1[i+1]*cos(E[0]*T[i+1]) - vY1[i+1]*sin(E[0]*T[i+1])
cY1[i+1] = vX1[i+1]*sin(E[0]*T[i+1]) + vY1[i+1]*cos(E[0]*T[i+1])

cX2[i+1] = vX2[i+1]*cos(E[1]*T[i+1]) - vY2[i+1]*sin(E[1]*T[i+1])
cY2[i+1] = vX2[i+1]*sin(E[1]*T[i+1]) + vY2[i+1]*cos(E[1]*T[i+1])

cX3[i+1] = vX3[i+1]*cos(E[2]*T[i+1]) - vY3[i+1]*sin(E[2]*T[i+1])
cY3[i+1] = vX3[i+1]*sin(E[2]*T[i+1]) + vY3[i+1]*cos(E[2]*T[i+1])

cX4[i+1] = vX4[i+1]*cos(E[3]*T[i+1]) - vY4[i+1]*sin(E[3]*T[i+1])
cY4[i+1] = vX4[i+1]*sin(E[3]*T[i+1]) + vY4[i+1]*cos(E[3]*T[i+1])

cX5[i+1] = vX5[i+1]*cos(E[4]*T[i+1]) - vY5[i+1]*sin(E[4]*T[i+1])
cY5[i+1] = vX5[i+1]*sin(E[4]*T[i+1]) + vY5[i+1]*cos(E[4]*T[i+1])

cX6[i+1] = vX6[i+1]*cos(E[5]*T[i+1]) - vY6[i+1]*sin(E[5]*T[i+1])
cY6[i+1] = vX6[i+1]*sin(E[5]*T[i+1]) + vY6[i+1]*cos(E[5]*T[i+1])

cX7[i+1] = vX7[i+1]*cos(E[6]*T[i+1]) - vY7[i+1]*sin(E[6]*T[i+1])
cY7[i+1] = vX7[i+1]*sin(E[6]*T[i+1]) + vY7[i+1]*cos(E[6]*T[i+1])

cX8[i+1] = vX8[i+1]*cos(E[7]*T[i+1]) - vY8[i+1]*sin(E[7]*T[i+1])

```

```

cY8[i+1] = vX8[i+1]*sin(E[7]*T[i+1]) + vY8[i+1]*cos(E[7]*T[i+1])

# CALCULO DE PROBABILIDADES POR CADA ESTADO

D1[i] = (cX1[i])**2 + (cY1[i])**2
D2[i] = (cX2[i])**2 + (cY2[i])**2
D3[i] = (cX3[i])**2 + (cY3[i])**2
D4[i] = (cX4[i])**2 + (cY4[i])**2
D5[i] = (cX5[i])**2 + (cY5[i])**2
D6[i] = (cX6[i])**2 + (cY6[i])**2
D7[i] = (cX7[i])**2 + (cY7[i])**2
D8[i] = (cX8[i])**2 + (cY8[i])**2

N[i] = D1[i] + D2[i] + D3[i] + D4[i] + D5[i] + D6[i] + D7[i] + D8[i]

D1[-1] = (cX1[-1])**2 + (cY1[-1])**2
D2[-1] = (cX2[-1])**2 + (cY2[-1])**2
D3[-1] = (cX3[-1])**2 + (cY3[-1])**2
D4[-1] = (cX4[-1])**2 + (cY4[-1])**2
D5[-1] = (cX5[-1])**2 + (cY5[-1])**2
D6[-1] = (cX6[-1])**2 + (cY6[-1])**2
D7[-1] = (cX7[-1])**2 + (cY7[-1])**2
D8[-1] = (cX8[-1])**2 + (cY8[-1])**2

N[-1] = D1[-1] + D2[-1] + D3[-1] + D4[-1] + D5[-1] + D6[-1] + D7[-1] +
D8[-1]

XMAX = np.array([max(cX1), max(cX2), max(cX3), max(cX4), max(cX5),
max(cX6), max(cX7), max(cX8)])
YMAX = np.array([max(cY1), max(cY2), max(cY3), max(cY4), max(cY5),
max(cY6), max(cY7), max(cY8)])
TMAX = max( np.concatenate( (XMAX,YMAX), axis = None ) ) + 0.1

XMIN = np.array([min(cX1), min(cX2), min(cX3), min(cX4), min(cX5),
min(cX6), min(cX7), min(cX8)])
YMIN = np.array([min(cY1), min(cY2), min(cY3), min(cY4), min(cY5),
min(cY6), min(cY7), min(cY8)])
TMIN = min( np.concatenate( (XMIN,YMIN), axis = None ) ) - 0.1

# GRAFICAS DE LA SOLUCION Y COEFICIENTES COMPLEJOS DE FUNCION DE ONDA.

sstyle = "seaborn-poster"
plt.style.use(sstyle)
plt.rc('text', usetex = True)
plt.rc('font', family = 'serif')

plt.figure(figsize = (20,10))

file_name = " X1_" + "%.3f"%vX1[0] + " Y1_" + "%.3f"%vY1[0] + "
X2_" + "%.3f"%vX2[0] + " Y2_" + "%.3f"%vY2[0] \
+ " X3_" + "%.3f"%vX3[0] + " Y3_" + "%.3f"%vY3[0] + "
X4_" + "%.3f"%vX4[0] + " Y4_" + "%.3f"%vY4[0] \
+ " X5_" + "%.3f"%vX5[0] + " Y5_" + "%.3f"%vY5[0] + "
X6_" + "%.3f"%vX6[0] + " Y6_" + "%.3f"%vY6[0] \

```

```

+ " X7_" + "%.3f"%vX7[0] + " Y7_" + "%.3f"%vY7[0] + "
X8_" + "%.3f"%vX8[0] + " Y8_" + "%.3f"%vY8[0] \
+ " w1_" + "%.3f"%w1 + " w2_" + "%.3f"%w2 + "
w3_" + "%.3f"%w3 + " J_" + "%.3f"%J \
+ " Jp_" + "%.3f"%Jp + " Ome_" + "%.3f"%U + "
ome_" + "%.3f"%w + " omega0_" + "%.3f"%w0 \
+ " phi_" + "%.3f"%phi + " h_" + "%.3f"%h + "_"

points = int(0.06*n)

plt.plot(T, N, label = "Total Probability", linewidth = 5.0 )
plt.plot(T, D1, label = "$|C_{000}(t)|^2$", marker="o", markevery=points)
plt.plot(T, D2, label = "$|C_{001}(t)|^2$", marker="v", markevery=points)
plt.plot(T, D3, label = "$|C_{010}(t)|^2$", marker="d", markevery=points)
plt.plot(T, D4, label = "$|C_{011}(t)|^2$", marker="p", markevery=points)
plt.plot(T, D5, label = "$|C_{100}(t)|^2$", marker="s", markevery=points)
plt.plot(T, D6, label = "$|C_{101}(t)|^2$", marker="X", markevery=points)
plt.plot(T, D7, label = "$|C_{110}(t)|^2$", marker="^", markevery=points)
plt.plot(T, D8, label = "$|C_{111}(t)|^2$", marker=".", markevery=points)

plt.vlines([tf1, tf2, tf3, tf4, tf5, tf6, tf7, tf8], 0, 1, colors='k')

plt.title( "Three-Qubit Probabilities of Quantum Teleportation")
plt.xlabel("Time (s)")
plt.ylabel("Probability")

plt.legend(loc=1)
plt.grid()

#plt.savefig( file_name + ".png", format = "png")
plt.show()

COMP = input( "Do you want to see the components of the wave function?
[Y/N]: ")

if(COMP == 'N' or COMP == 'n' or COMP == 'No' or COMP == 'NO' or COMP ==
'no' or COMP == 'Nope' or COMP == 'nope'):
    pass

elif (COMP == 'Y' or COMP == 'y' or COMP == 'Yes' or COMP == 'YES' or
COMP == 'yes'):

    # COEFICIENTES DE LA FUNCION DE ONDA
    # -----
    # PRIMER COEFICIENTE
    # -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX1, label = "$X_{1}$ - Real ")
plt.plot(T, cY1, label = "$Y_{1}$ - Imaginary")
plt.title( "Components of the Coefficient $C_{000}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)

```

```

plt.grid()
plt.show()

# -----
# SEGUNDO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX2, label = "$X_{2}$ - Real ")
plt.plot(T, cY2, label = "$Y_{2}$ - Imaginary")
plt.title( "Components of the Coeficiente $C_{001}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# TERCER COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX3, label = "$X_{3}$ - Real ")
plt.plot(T, cY3, label = "$Y_{3}$ - Imaginary")
plt.title( "Components of the Coeficiente $C_{010}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# CUARTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX4, label = "$X_{4}$ - Real ")
plt.plot(T, cY4, label = "$Y_{4}$ - Imaginary")
plt.title( "Components of the Coeficiente $C_{011}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# QUINTO COEFICIENTE

```

```

# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX5, label = "$X_{5}$ - Real ")
plt.plot(T, cY5, label = "$Y_{5}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{100}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEXTO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX6, label = "$X_{6}$ - Real ")
plt.plot(T, cY6, label = "$Y_{6}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{101}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# SEPTIMO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

plt.plot(T, cX7, label = "$X_{7}$ - Real ")
plt.plot(T, cY7, label = "$Y_{7}$ - Imaginary")
plt.title( "Components of the Coeficient $C_{110}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()

# -----
# OCTAVO COEFICIENTE
# -----

plt.figure(figsize = (20,10))
plt.ylim(top = TMAX)
plt.ylim(bottom = TMIN)

```

```
plt.plot(T, cX8, label = "$X_{8}$ - Real ")
plt.plot(T, cY8, label = "$Y_{8}$ - Imaginary")
plt.title( "Components of the Coefficient $C_{111}(t)$")
plt.xlabel("Time (s)")
plt.ylabel(" ")
plt.legend(loc=0)
plt.grid()
plt.show()
```