

Relatividad General

Modelo de evolución de un pulsar binario

Giovanni Gamaliel López Padilla, Ivan Arturo Pla Gúzman

5 de noviembre de 2020

Introducción - Punto de vista newtoniano

La mecánica newtoniana es un modelo físico que funciona para describir la dinámica de los cuerpos en el espacio por medio de las fuerzas que contiene cada objeto. Históricamente, la mecánica newtoniana fue el primer modelo físico en poder representar de buena manera la dinámica de los objetos al punto de predecir acciones importantes sobre el movimiento de los cuerpos, en donde se incluyen las trayectorias de ciertos planetas.

$$\dot{\mathbf{p}} = \frac{d(m\mathbf{v})}{dt} = \mathbf{F}, \quad \left\{ \begin{array}{l} \dot{p}_x = \frac{d(m_x)}{dt} = F_x \\ \dot{p}_y = \frac{d(m_y)}{dt} = F_y \\ \dot{p}_z = \frac{d(m_z)}{dt} = F_z \end{array} \right.$$

El problema físico clásico puede plantearse de forma simplificada como:

Dadas las propiedades orbitales (masa, posición instantánea y velocidad) de un grupo de cuerpos astronómicos, determinar las fuerzas interactivas actuantes; y consiguientemente, calcular sus movimientos orbitales para cualquier instante futuro.

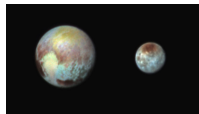
Introducción - Sistemas binarios

En astronomía, el término sistema binario se utiliza para referirse a dos objetos astronómicos que orbitan alrededor de un centro de masa común debido a la fuerza gravitatoria que hay dada su cercanía. Normalmente se utiliza para referirse a dos estrellas, sin embargo, el término puede aplicarse a un sistema formado por un planeta y un satélite natural, siempre y cuando este último sea excepcionalmente grande en comparación con el planeta. Algunos ejemplos de estos son:

- La galaxia de Bode (M81) y la galaxia del Cigarro (M82).
- Dentro del sistema solar, el sistema Plutón-Caronte está formado por un planeta y un satélite.



(a) Galaxia de Bode



(b) Plutón y caronte

Figura 1: Ejemplos de sistemas binarios

Introducción - Origen y principios de la relatividad general

Las características esenciales de la teoría de la relatividad general son las siguientes:

- El principio general de covariancia: las leyes de la Física deben tomar la misma forma matemática en todos los sistemas de coordenadas.
- El principio de equivalencia o de invariancia local de Lorentz: las leyes de la relatividad especial (espacio plano de Minkowski) se aplican localmente para todos los observadores inerciales.
- La curvatura del espacio-tiempo es lo que observamos como un campo gravitatorio, en presencia de materia la geometría del espacio-tiempo no es plana sino curva, una partícula en movimiento libre inercial en el seno de un campo gravitatorio sigue una trayectoria geodésica.

Objetivo

- Obtener la solución analítica que describa la evolución del semieje mayor a y la excentricidad e de un sistema pulsar binario.
- Obtener la solución numérica que describa la evolución del semieje mayor a y la excentricidad e de un sistema pulsar binario. Realizar una comparación de la solución analítica y la solución numérica con los valores observaciones reportados por Weisberg en 2010.

Definición

Un bit cuántico, qbit o qubit es la unidad básica de información cuántica, es la versión cuántica del clásico bit binario.

Estados basicos del qubit.

$|0\rangle$

$|1\rangle$

Por convención se llega a utilizar la siguiente expresión:

$|1100\rangle \rightarrow |12\rangle$

Marco teórico - Compuertas lógicas clásicas y cuánticas - Compuerta NOT

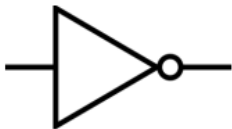


Figura 2: Representación gráfica de la compuerta NOT.

A	NOT(A)
0	1
1	0

Tabla 1: Compuerta clásica NOT.

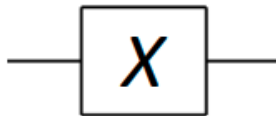


Figura 3: Representación gráfica de la compuerta cuántica NOT.

A	NOT(A)
$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$
$ \psi\rangle$	$\hat{X} \psi\rangle$

Tabla 2: Compuerta cuántica NOT.

Compuertas lógicas clásicas y cuánticas - Compuerta de Hadamard

$$|0\rangle \xrightarrow{\hat{H}} \frac{|0\rangle + |1\rangle}{\sqrt{2}} \equiv |+\rangle$$

$$|1\rangle \xrightarrow{\hat{H}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \equiv |-\rangle$$

donde:

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

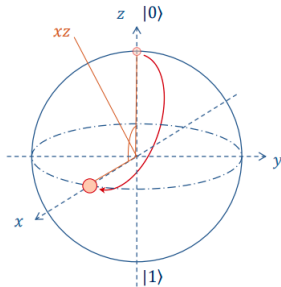


Figura 4: Representación gráfica del efecto de la transformación de Hadamard aplicada a un qubit.

Compuertas lógicas clásicas y cuánticas - Compuerta de desplazamiento de fase

$$|0\rangle \xrightarrow{\hat{R}(\phi)} |0\rangle$$

$$|1\rangle \xrightarrow{\hat{R}(\phi)} e^{i\phi} |1\rangle$$

donde:

$$\hat{R}(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}$$

Compuertas lógicas clásicas y cuánticas - Compuerta SWAP

$$U_{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

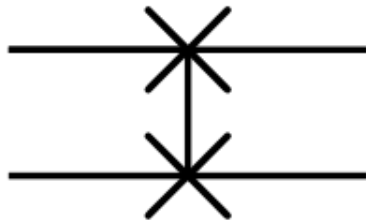


Figura 5: Representación gráfica de la compuerta SWAP.

Compuertas lógicas clásicas y cuánticas - Compuerta Controlada

$$U_{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

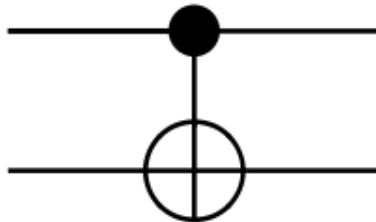
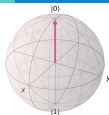


Figura 6: Representación gráfica de la compuerta NOT controlada.

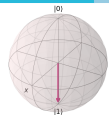
$$|\tilde{x}\rangle \equiv QFT |x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i xy}{N}} |y\rangle$$

$$|\tilde{x}\rangle = \frac{1}{\sqrt{N}} \bigotimes_{k=1}^N \left[|0\rangle + e^{2\pi i x 2^{-k}} |1\rangle \right].$$

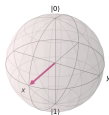
Transformada de Fourier Cuántica



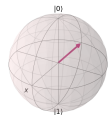
(a) Qubit con el estado cuántico $|0\rangle$ representado en la esfera de Bloch.



(b) Qubit con el estado cuántico $|1\rangle$ representado en la esfera de Bloch.



(c) Qubit con el estado cuántico $|\tilde{0}\rangle$ representado en la esfera de Bloch.



(d) Qubit con el estado cuántico $|\tilde{1}\rangle$ representado en la esfera de Bloch.

Figura 7: Representación en la esfera de Bloch de los estados bases computacionales cuánticos y las bases de Fourier.

Estimación de Fase Cuántica (QPE)

$$\mathcal{U} |\psi\rangle = e^{i\theta_\psi} |\psi\rangle .$$

Periodo de la función $a^x \bmod N$

$$f(x) = a^x \bmod N \quad (1)$$

$$a^r \bmod N = 1 \quad (2)$$

Usando como ejemplo $a=3$ y $N=35$, se tiene el siguiente periodo:

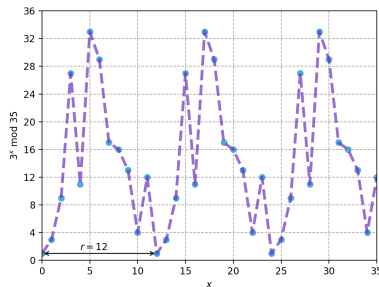


Figura 8: Periodo de la función 1 para visualizar la condición 2

Periodo de la función $a^x \bmod N$

$$\mathcal{U} |1\rangle = |3\rangle$$

$$\mathcal{U}^2 |1\rangle = |9\rangle$$

$$\mathcal{U}^3 |1\rangle = |12\rangle$$

$$\vdots$$

$$\mathcal{U}^{r-1} |1\rangle = |12\rangle$$

$$\mathcal{U}^r |1\rangle = |1\rangle$$

$$|u_0\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |a^k \bmod N\rangle. \quad (3)$$

Periodo de la función $a^x \bmod N$

$$\begin{aligned} |u_0\rangle &= \frac{1}{\sqrt{12}} (|1\rangle + |3\rangle + |9\rangle + \cdots + |4\rangle + |12\rangle) \\ U|u_0\rangle &= \frac{1}{\sqrt{12}} (U|1\rangle + U|3\rangle + U|9\rangle + \cdots + U|4\rangle + U|12\rangle) \\ &= \frac{1}{\sqrt{12}} (|3\rangle + |9\rangle + |27\rangle + \cdots + |12\rangle + |1\rangle) \\ &= |u_0\rangle. \end{aligned}$$

Periodo de la función $a^x \bmod N$

$$|u_1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i k}{r}} |a^k \bmod N\rangle$$

$$U|u_1\rangle = e^{\frac{2\pi i}{r}} |u_1\rangle.$$

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle.$$

Periodo de la función $a^x \bmod N$

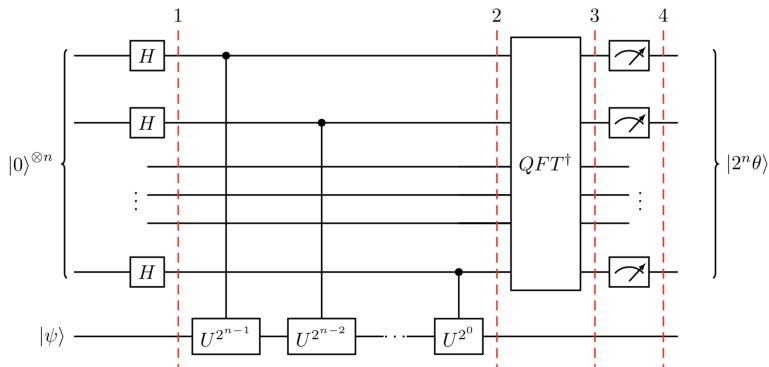


Figura 9: Circuito cuántico que realiza el algoritmo QPE (Quantum Phase Estimation) sobre una serie de qubits.

Implementación en Qiskit

```
In [2]: from qiskit import QuantumCircuit

def a_x_mod15(a, x):
    if a not in [2,7,8,11,13]:
        raise ValueError("'a' must be 2,7,8,11 or 13")
    U = QuantumCircuit(4)
    for iteration in range(x):
        if a in [2,13]:
            U.swap(0,1)
            U.swap(1,2)
            U.swap(2,3)
        if a in [7,8]:
            U.swap(2,3)
            U.swap(1,2)
            U.swap(0,1)
        if a == 11:
            U.swap(1,3)
            U.swap(0,2)
        if a in [7,11,13]:
            for q in range(4):
                U.x(q)
    U = U.to_gate()
    U.name = "%i^%i mod 15" % (a, x)
    c_U = U.control()
    return c_U
```

Implementación en Qiskit

```
In [3]: def modular_exponentiation(given_circuit, n, m, a):  
        for x in range(n):  
            exponent = 2**x  
            given_circuit.append(a_x_mod15(a, exponent),  
                                [x] + list(range(n, n+m)))
```

```
In [37]: def shor_program(n, m, a):  
        # set up quantum circuit  
        shor = QuantumCircuit(n+m, n)  
        # initialize the qubits  
        initialize_qubits(shor, n, m)  
        shor.barrier()  
        # apply modular exponentiation  
        modular_exponentiation(shor, n, m, a)  
        shor.barrier()  
        # apply inverse QFT  
        apply_iqft(shor, range(n))  
        # measure the first n qubits  
        shor.measure(range(n), range(n))  
        return shor  
  
        n = 4; m = 4; a = 7  
        mycircuit = shor_program(n, m, a)  
        mycircuit.draw(output='text')
```

Implementación con Qiskit

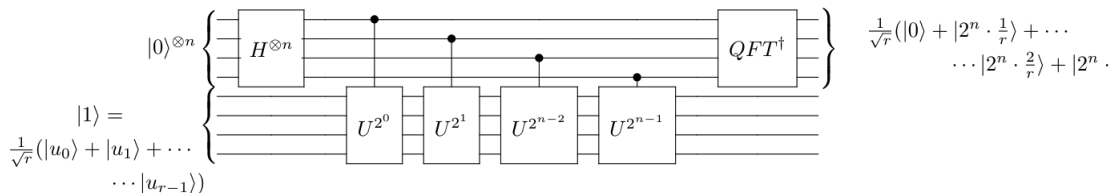


Figura 10: Algoritmo de Shor para la factorización de un número visto desde la estructura que propone Qiskit.

Algoritmo	Número a factorizar	IBM Device	Lanzamientos	Repeticiones
CCGN	21	-	-	1000
Shor (Local)		Vigo	1000	
Shor (IBM Q)				

Tabla 3: Parámetros de entrada de cada algoritmo de factorización.

Algoritmo	Promedio (ms)	σ (ms)
Clasico	0.056	0.0096
Shor	0.036	0.0153
Shor IBM	0.024	0.0102

Tabla 4: Promedio y desviación estandar de cada algoritmo de factotización ejecutados en una computadora clásica y cuántica.

Resultados

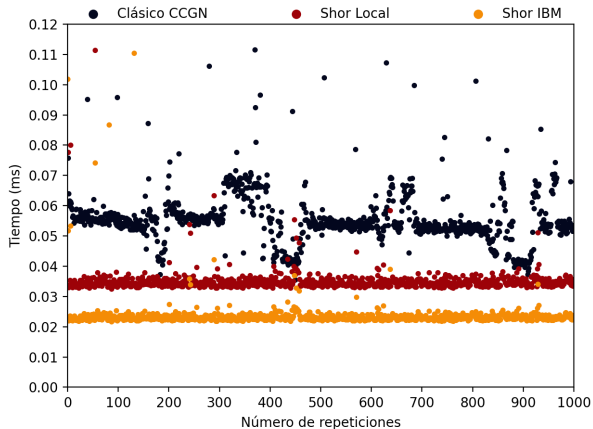


Figura 11: Comparación de los tiempos de procesamiento de los códigos ejecutados en una computadora clásica y en una computadora cuántica.

Conclusiones y discusión

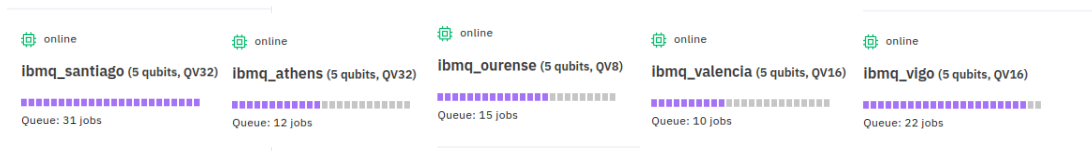


Figura 12: Lista de dispositivos disponibles al público desde *IBM Q Experience*

Referencias

Agrios. *Introducción a La Teoría de Números.*, volume 3. 2003.

G. P. Berman, G. D. Doolen, G. V. López, and V. I. Tsifrinovich. Nonresonant effects in the implementation of the quantum Shor algorithm. *Physical Review A - Atomic, Molecular, and Optical Physics*, 61(4):7, 2000.

Vicente Moret Bonillo. Principios Fundamentales De Computación Cuántica. *universidad de A Coruña*, pages 1–181, 2013.

Luanne S. Cohen and Tanya Wendling. Técnicas de diseño. *Técnicas de diseño*, pages 15–18, 1998.

Edward Gerjuoy. Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers. *American Journal of Physics*, 73(6):521–540, 2005.

F. Ghisi and S. V. Ulyanov. The information role of entanglement and interference operators in Shor quantum algorithm gate dynamics. *Journal of Modern Optics*, 47(12):2079–2090, 2000.

Daniel Koch, Saahil Patel, Laura Wessing, and Paul M. Alsing. Fundamentals In Quantum Algorithms: A Tutorial Series Using Qiskit Continued. 2020.

Samuel J. Lomonaco and Louis H. Kauffman. A continuous variable Shor algorithm. pages 97–108, 2005.

Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

D E Ingenieros D E Telecomunicación. Números primos especiales y sus aplicaciones criptográficas. 2003.

Lieven M.K. Vandersypen, Matthias Breyta, Gregory Steffen, Costantino S. Yannoni, Mark H. Sherwood, and Isaac L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(6866):883–887, 2001.

Anocha Yimsiriwattana and Samuel J. Lomonaco Jr. Distributed quantum computing: a distributed Shor algorithm. *Quantum Information and Computation II*, 5436:360, 2004.

S. S. Zhou, T. Loke, J. A. Izaac, and J. B. Wang. Quantum Fourier transform in computational basis. *Quantum Information Processing*, 16(3):1–19, 2017.