

Formalismo de Matriz Densidad e Integrales de Camino para una partícula en un Potencial de interacción mediante técnicas de Monte Carlo

Laura Catalina Arboleda Hernández*

*Instituto de Física, Universidad de Antioquia

Resumen

Se desarrolló el método de *matrix-squaring* mediante el uso de la propiedad de convolución de matriz densidad para estimar la densidad de probabilidad para una partícula en un potencial armónico y uno anarmónico. Se realizó un muestreo de Monte Carlo para evaluar la aproximación a la integral de camino de Feynman. Los caminos individuales se definen sobre una malla de tiempo imaginaria, en este caso en una dimensión (una recta discretizada en nodos) con condiciones de frontera periódicas, es decir, caminos individuales cerrados. En cada nodo del camino, se especifica la ubicación espacial de la partícula a través de la variable de posición x_i . Usando el algoritmo de Metropolis se reproduce la densidad de probabilidad de ocurrencia de cada uno de esos caminos, con pesos dados por la matriz de densidad de la partícula sometida a un potencial de interacción en la aproximación de Trotter. Se obtuvo la forma exacta de la distribución de probabilidad de los osciladores armónicos cuántico y clásico a altas temperaturas a partir de la expansión de Trotter y se compararon estas distribuciones exactas con el muestreo obtenido con el muestreo de Monte Carlo.

Palabras Clave

Matriz densidad, función partición, propiedad de convolución, cadenas de Markov.

1. Introducción

A través de las integrales de camino de Feynman se puede encontrar una conexión entre la física estadística y la teoría cuántica. Así, en un formalismo de tiempo imaginario una

integral de camino de Feynman es matemáticamente equivalente a la función partición.

La integral de camino es en esencia una suma sobre todas las posibles líneas de mundo que conectan las posiciones iniciales y finales. Una forma de sumar sobre todos los “posibles caminos” es introduciendo una malla temporal, de modo que los distintos caminos pueden ser etiquetados individualmente cuando el eje de tiempo se discretiza, así

$$x(t_k) = x_k; k = 0, 1, \dots, N. \quad (1)$$

con $-\infty < x_k < \infty$ para todos los nodos k . Cada camino será pesado por un factor exponencial “amortiguado” por los valores de la posición x_k , de modo que cada camino tiene una posibilidad de ocurrencia distinta, que hará que unos caminos contribuyan más que otros y viceversa. La función partición puede entonces estimarse como la suma de todos estos posibles caminos:

$$Z = \int \int dx_0, \dots, dx_{N-1} \rho(x_0, x_1, \Delta\tau) \dots \rho(x_{N-1}, x_0, \Delta\tau) \quad (2)$$

donde los elementos de matriz densidad representan los pesos con que contribuye cada camino. De esta forma tenemos una malla unidimensional con nodos etiquetados por el índice k . En cada nodo hay una variable x_k que toma un valor continuo. La integral $\int \int dx_0, \dots, dx_{N-1}$ es realmente una suma sobre todos los posibles configuraciones de los nodos. El movimiento de un nodo da como resultado un nuevo camino. El objetivo es evaluar las integrales que aparecen en la función partición. Aunque para el caso del oscilador armónico existe una solución exacta, en general la mayoría de veces está solución no existe. N es el número de integraciones que se realizan y en el límite $N \rightarrow \infty$ los errores en la aproximación son despreciables. En la sección 2 se hace la deducción de las curvas teóricas de probabilidad a partir de la expansión de Trotter y de la propiedad de convolución de la matriz densidad. En la sección 3.1 se estiman estas probabilidades usando la propiedad de convolución en un esquema implícito, para un potencial armónico. En la sección 3.2 se presenta una aproximación para la evaluación de la suma sobre todos los posibles caminos en la integral de Feynman basada en una técnica de Monte Carlo, concretamente en el algoritmo Metropolis. Además, se aplican ambos métodos pero para un potencial anarmónico. Finalmente, en la sección 4 se presentan los resultados obtenidos con los códigos implementados y una discusión de los mismos.

2. Oscilador Armónico Cuántico

Para el caso de una partícula cuántica sometida a un potencial armónico, usaremos la expansión de Trotter, en el límite altas temperaturas, para encontrar el elemento de matriz densidad en la base de posición: $\rho(x, x', \beta)$ y con él encontrar

la probabilidad de encontrar la partícula en una posición dada o la probabilidad de transición de una posición x a una posición x' .

La expansión de Trotter es válida sólo a altas temperaturas ($\beta \rightarrow 0$) y está dada por (Krauth [2006])

$$\rho(x, x', \beta) = e^{-\beta \frac{V(x)}{2}} \rho(x, x', \beta)^{free} e^{-\beta \frac{V(x')}{2}} \quad (3)$$

donde $\rho(x, x', \beta)^{free}$ corresponde al elemento de matriz densidad en la base de posición de una partícula libre (sin término de interacción), el cuál está dado por

$$\rho(x, x', \beta)^{free} = \sum_{n=0}^{\infty} \Psi_n^*(x) e^{-\beta E_n} \Psi_n(x') \quad (4)$$

Para una partícula libre, usando unidades reducidas $\hbar = m = \omega = 1$

$$\hat{H} = \frac{\hat{P}^2}{2m} = -\frac{1}{2} \frac{\partial^2}{\partial x^2} \Psi_n(x) \quad (5)$$

siendo la función de onda para el oscilador en una dimensión $\Psi_n(x) = (1/\sqrt{L}) e^{ik_x x}$, con condiciones de frontera periódicas y $k_x = 2\pi n_x/L$. La energía está dada entonces por

$$E_n = \frac{\hbar^2 k_x^2}{2m} = \frac{k_x^2}{2} \quad (6)$$

Sustituyendo esto en la ecuación 4, se obtiene

$$\begin{aligned} \rho(x, x', \beta)^{free} &= \frac{1}{L} \sum_{n=0}^{\infty} e^{-ik_x x} e^{-\beta \frac{k_x^2}{2}} e^{ik_x x'} \\ &= \frac{1}{L} \sum_{n=0}^{\infty} e^{-ik_x (x-x')} e^{-\beta \frac{k_x^2}{2}} \end{aligned} \quad (7)$$

Para longitudes grandes, la separación entre los números de onda k_x es muy pequeña, es decir que se puede hablar de un continuo:

$$\lim_{\Delta k \rightarrow 0} \sum f(k) \Delta k \approx \int f(k) dk$$

Dado que los n_x son enteros, entonces $\Delta k = 2\pi/L$, por lo tanto la ecuación 7 se escribe como

$$\begin{aligned} \rho(x, x', \beta)^{free} &= \frac{1}{2\pi} \int_0^\infty e^{-ik_x (x-x')} e^{-\beta \frac{k_x^2}{2}} dk \\ &= \frac{1}{2\pi} \int_0^\infty e^{-\left(ik(x-x') + \beta \frac{k^2}{2}\right)} dk \end{aligned} \quad (8)$$

para completar cuadrados en el exponente de la exponencial en la ecuación 8, vemos que

$$\left(\frac{i(x-x')}{\sqrt{2\beta}} + k_x \sqrt{\frac{\beta}{2}} \right)^2 = -\frac{(x-x')^2}{2\beta} + ik_x (x-x') + k_x^2 \frac{\beta}{2}$$

Luego, sumando y restando el término $\frac{(x-x')^2}{2\beta}$ en ese exponente (8), se obtiene

$$\rho(x, x', \beta)^{free} = \frac{1}{2\pi} e^{-\frac{(x-x')^2}{2\beta}} \int_0^\infty e^{-\left(\frac{i(x-x')}{\sqrt{2\beta}} + k_x \sqrt{\frac{\beta}{2}}\right)^2} dk \quad (9)$$

La integral en 9 es una integral gaussiana (ecuación ??), con lo que se obtiene finalmente el elemento de matriz densidad para una partícula libre en la base de posición

$$\rho(x, x', \beta)^{free} = \frac{1}{\sqrt{2\pi\beta}} e^{-\frac{(x-x')^2}{2\beta}} \quad (10)$$

Volviendo a la expansión de Trotter y reemplazando 10 en 3, para un potencial armónico $V(x) = x^2/2$, se tiene

$$\begin{aligned} \rho(x, x', \beta) &= \frac{1}{\sqrt{2\pi\beta}} e^{-\beta \frac{x^2}{4}} e^{-\frac{(x-x')^2}{2\beta}} e^{-\beta \frac{x'^2}{4}} \\ &= \frac{1}{\sqrt{2\pi\beta}} e^{-\left(\frac{(x-x')^2}{2\beta} + \frac{\beta}{4}(x^2+x'^2)\right)} \\ &= \frac{1}{\sqrt{2\pi\beta}} e^{-\left(\frac{(x-x')^2}{2\beta} + \frac{\beta}{4}\frac{(x-x)^2 + (x+x')^2}{2}\right)} \\ &= \frac{1}{\sqrt{2\pi\beta}} e^{-\left(\left[\frac{1}{\beta} + \frac{\beta}{4}\right]\frac{(x-x')^2}{2} + \frac{\beta}{4}\frac{(x+x')^2}{2}\right)} \end{aligned} \quad (11)$$

llamaremos

$$\begin{aligned} c(\beta) &= \frac{1}{\sqrt{2\pi\beta}} \\ g(\beta) &= \frac{1}{\beta} + \frac{\beta}{4} \\ f(\beta) &= \frac{\beta}{4} \end{aligned} \quad (12)$$

así, reescribiendo 11 en términos de la funciones anteriores

$$\rho(x, x', \beta) = c(\beta) e^{-g(\beta) \frac{(x-x')^2}{2}} e^{-f(\beta) \frac{(x+x')^2}{2}} \quad (13)$$

Caso particular: oscilador armónico clásico ($T \rightarrow 0$)

En este punto, usando la ecuación 11 en el límite de bajas temperaturas podemos obtener la probabilidad $\Pi(x)_{clasica}$ de que una partícula clásica esté en la posición x

$$\begin{aligned} \Pi(x)_{clasica} &= \frac{\rho(x, x, \beta)}{\mathcal{Z}} \\ &= \frac{\rho(x, x, \beta)}{\int \rho(x, x, \beta) dx} \end{aligned} \quad (14)$$

El elemento de matriz $\rho(x, x, \beta)$ puede obtenerse de la ecuación 11 haciendo $x = x'$. Se obtiene

$$\rho(x, x, \beta) = \frac{1}{\sqrt{2\pi\beta}} e^{-\beta \frac{x^2}{2}}$$

con este resultado, la función de partición es

$$\begin{aligned}\mathcal{Z} &= \int \rho(x, x, \beta) dx \\ &= \frac{1}{\sqrt{2\pi\beta}} \int_{-\infty}^{\infty} e^{-\beta \frac{x^2}{2}} dx \\ &= \frac{1}{\beta}\end{aligned}\quad (15)$$

Así, la probabilidad $\Pi(x)_{clasica}$ es

$$\Pi(x)_{clasica} = \sqrt{\frac{\beta}{2\pi}} e^{-\beta \frac{x^2}{2}}$$

Probabilidad Cuántica:

Dada la propiedad de convolución de la matriz densidad:

$$\rho(x, x'', \beta_1 + \beta_2) = \int \rho(x, x', \beta_1) \rho(x', x'', \beta_2) dx' \quad (16)$$

Si $\beta_1 = \beta_2 = \beta$, y usando la ecuación 13, la ecuación 16 se convierte en

$$\begin{aligned}\rho(x, x'', 2\beta) &= c^2 \int e^{\left[g \frac{(x-x')^2}{2} - f \frac{(x+x')^2}{2} - g \frac{(x'-x'')^2}{2} - f \frac{(x'+x'')^2}{2} \right]} dx' \text{ Veamos que} \\ &= c^2 \int e^{-\frac{g}{2} [(x-x')^2 + (x'-x'')^2] - \frac{f}{2} [(x+x')^2 + (x'+x'')^2]} dx' \\ &\quad (17)\end{aligned}$$

Para el exponente de la exponencial en la ecuación anterior se tiene:

$$\begin{aligned}&= -\frac{g}{2} [(x-x')^2 + (x'-x'')^2] - \frac{f}{2} [(x+x')^2 + (x'+x'')^2] \\ &= -\frac{g}{2} (x^2 + x''^2 + 2x'x - 2x'x'') \\ &\quad - \frac{f}{2} (x^2 + x''^2 + 2x'x' + 2x'x'') \\ &= -\frac{g+f}{2} (x^2 + x''^2) - (g+f)x'x - (f-g)(x+x'')x' \\ &\quad (19)\end{aligned}\quad (18)$$

Luego, la ecuación 17 se convierte en

$$\rho(x, x'', 2\beta) = c^2 e^{-\frac{(g+f)}{2} (x^2 + x''^2)} \times \quad (20)$$

$$\int e^{-[(g+f)x'^2 + (f-g)(x+x'')x']} dx' \quad (21)$$

De nuevo completando cuadrados en el exponente de la función en el integrando y resolviendo la integral gaussiana, se obtiene

$$\rho(x, x'', 2\beta) = c^2 e^{-\frac{(g+f)}{2} (x^2 + x''^2)} e^{-\frac{(f-g)^2 (x+x'')^2}{4(g+f)}} \quad (22)$$

$$\begin{aligned}&\int e^{-\left[\sqrt{(g+f)}x' + \frac{(f-g)(x+x'')}{2\sqrt{(g+f)}} \right]^2} dx' \\ &= c^2 e^{-\frac{(g+f)}{2} (x^2 + x''^2)} e^{-\frac{(f-g)^2 (x+x'')^2}{4(g+f)}}\end{aligned}\quad (23)$$

$$\sqrt{\frac{\pi}{g+f}} \quad (24)$$

Llamando

$$c(2\beta) = c^2 \sqrt{\frac{\pi}{g+f}}$$

entonces

$$\rho(x, x'', 2\beta) = c(2\beta) e^{-\frac{(x+x'')^2}{2} f(2\beta) - \frac{(x-x'')^2}{2} g(2\beta)} \quad (25)$$

en donde $f(2\beta)$ y $g(2\beta)$ se obtienen usando las relaciones en 12

$$\begin{aligned}f(2\beta) &= \left[\frac{f+g}{2} - \frac{(f-g)^2}{2(f+g)} \right] \\ g(2\beta) &= \frac{f+g}{2}\end{aligned}\quad (26)$$

$$g(2\beta) = \frac{f+g}{2} = \frac{\frac{1}{4g} + g}{2} \quad (27)$$

La función que satisface 27 es

$$g(\beta) = \frac{1}{2} \coth(\beta/2) = \frac{1}{2} \frac{1}{\tanh(\beta/2)} \quad (28)$$

Así, como

$$f(\beta)g(\beta) = \frac{\beta}{4} \left(\frac{1}{\beta} + \frac{\beta}{4} \right) = \frac{1}{4} \quad (29)$$

entonces

$$f(\beta) = \frac{1}{4g(\beta)} = \frac{1}{2} \tanh(\beta/2) \quad (30)$$

$$c(2\beta) = \sqrt{\frac{1}{2\pi \sinh(2\beta)}} \quad (31)$$

De esta forma, conociendo $c(\beta)$, $g(\beta)$ y $f(\beta)$, podemos volver a la ecuación 13 y reescribirla,

$$\begin{aligned}\rho(x, x', \beta) &= c(\beta) e^{-g(\beta) \frac{(x-x')^2}{2} - f(\beta) \frac{(x+x')^2}{2}} \\ &= \frac{1}{\sqrt{2\pi \sinh(\beta)}} e^{-\frac{(x-x')^2}{4} \coth(\frac{\beta}{2}) - \frac{(x+x')^2}{4} \tanh(\frac{\beta}{2})}\end{aligned}\quad (32)$$

Los elementos de la diagonal ($x = x'$) son

$$\rho(x, x, \beta) = \frac{1}{\sqrt{2\pi \sinh(\beta)}} e^{-x^2 \tanh(\beta/2)} \quad (33)$$

Con esto, la probabilidad de que la partícula cuántica se encuentre en la posición x está dada por

$$\begin{aligned} \Pi(x)_{cuantica} &= \frac{\rho(x, x, \beta)}{\mathcal{Z}} \\ &= \frac{\rho(x, x, \beta)}{\int \rho(x, x, \beta) dx} \\ &= \frac{e^{-x^2 \tanh(\beta/2)}}{\int_{-\infty}^{\infty} e^{-x^2 \tanh(\beta/2)} dx} \\ &= \frac{e^{-x^2 \tanh(\beta/2)}}{\sqrt{\frac{\pi}{\tanh(\beta/2)}}} \\ &= \sqrt{\frac{\tanh(\beta/2)}{\pi}} e^{-x^2 \tanh(\beta/2)} \end{aligned} \quad (34)$$

3. Oscilador Armónico Cuántico

3.1. Matrix-Squaring

La distribución de probabilidad para una partícula sometida a un potencial armónico, en contacto con un baño térmico está dada por

$$\Pi(x) = \frac{\rho(x, x, \beta)}{\mathcal{Z}} \quad (35)$$

Usaremos el método de matrix squaring para estimar la matriz densidad. Como se mencionó en la sección 2, apartir de la matriz densidad con temperatura inversa β , se puede estimar la matriz densidad a una temperatura inversa de 2β . De esta forma, se implementó un código en Python para obtener $\rho(x, x, 2\beta)$ a partir de $\rho(x, x, \beta)$, así

$$\rho(x, x'', 2\beta) = \int dx' \rho(x, x', \beta) \rho(x', x'', \beta) \quad (36)$$

La integral se puede aproximar como una suma, luego

$$\rho(x, x', 2\beta) \approx \sum_k \rho(x, x_k, \beta) \rho(x_k, x', \beta) \Delta x_k \quad (37)$$

como nos interesan los elementos de la diagonal, $x = x'$, entonces

$$\rho(x, x, 2\beta) \approx \sum_k \rho(x, x_k, \beta) \rho(x_k, x, \beta) \Delta x_k \quad (38)$$

Una vez conocida la matriz densidad, se puede obtener la función partición aproximando de nuevo la integral como una suma

$$\mathcal{Z} = \int \rho(x, x, 2\beta) dx \approx \sum \rho(x, x, 2\beta) \Delta x \quad (39)$$

Conocido $\rho(x, x, \beta)$ y \mathcal{Z} se puede obtener la distribución de probabilidad de la ecuación 35. Dado que la estimación con la propiedad de convolución de la matriz densidad depende fuertemente de los valores iniciales del inverso de la temperatura β_0 y el paso Δx , se estimó $\Pi(x)$ variando estos dos parámetros para ver qué combinación de ellos arrojaba los mejores resultados. El código implementado se muestra a continuación.

```
#Elementos no diagonales de la
#matriz densidad de una partícula libre

def rho_free(x, xp, beta):
    return (np.exp(-(x-xp)**2/(2.0*beta))\
            /np.sqrt(2.0*np.pi*beta))

#matriz densidad oscilador armónico
#en la aproximación de Trotter

def rho_harmonic_trotter(grid, beta):
    return np.array([rho_free(x, xp, beta)*\
                    np.exp(-0.5*beta*0.5*(x**2 + xp**2)) \
                    for x in grid] for xp in grid])

#matriz densidad para 2beta

def density_matrix_2b(x, beta_tmp, beta, dx, nx):

    #matriz densidad a beta inicial
    rho = rho_harmonic_trotter(x, beta_tmp)

    while beta_tmp < beta:
        rho = np.dot(rho, rho)
        rho *= dx
        beta_tmp *= 2.0

    #función partición
    Z = sum(rho[j, j] for j in range(nx)) * dx

    #Probabilidad de estar: x=x'
    pix = [rho[j, j]/Z for j in range(nx)]

    return pix, beta_tmp

#definición de probabilidades exactas

def quantum_probability(x, b):
    Pi_quantum = np.sqrt(np.tanh(b/2.0)/np.pi)*\
                np.exp(-(x**2)*(np.tanh(b/2.0)))
    return Pi_quantum

def classic_probability(x, b):
    Pi_classic = np.sqrt(b/(2*np.pi))*\
                np.exp(-(b*(x**2))/2.0)
    return Pi_classic

#grafica y escritura de archivo de datos
```

```

x_max = 5.0
nx = 100

xa = int(-(nx-1)/2)
xb = int(nx/2+1)

#valor inicial de beta
beta_tmp = 2**(-7)

#valor real de beta: target: beta=4
beta = 2.0**(2)

dx = [0.01,0.1,0.2,1.0]

for k in dx:

    x = [i*k for i in range(xa, xb)]

    for p in beta_tmp:

        pi_of_x,beta_tmp1 = \
            density_matrix_2b(x,p,beta,k,nx)

        #escribir archivo de datos..

        f = open('datos.dat','w')

        for j in range (nx):
            f.write("%f %f\n"%(x[j],pi_of_x[j]))
        f.close()

        b = beta_tmp1

        y_quantum = [quantum_probability(m,b)\
                     for m in x]

        y_classic = [classic_probability(m,b)\
                     for m in x]

        #gráficas teóricas

        plt.plot(x, y_quantum)
        plt.plot(x, y_classic)

        #gráfica datos

        plt.plot(x, pi_of_x)

```

3.1.1. Determinación valores iniciales ($\beta_0, \Delta x$)

Con el código descrito en la sección anterior se estimó la distribución de probabilidad ($\Pi(x)$) para la partícula, de encontrarse en una cierta posición x cuando está sometida a un potencial armónico (ecuación 35), para distintos parámetros iniciales. Para obtener $\rho(x, x, \beta = 4)$, en

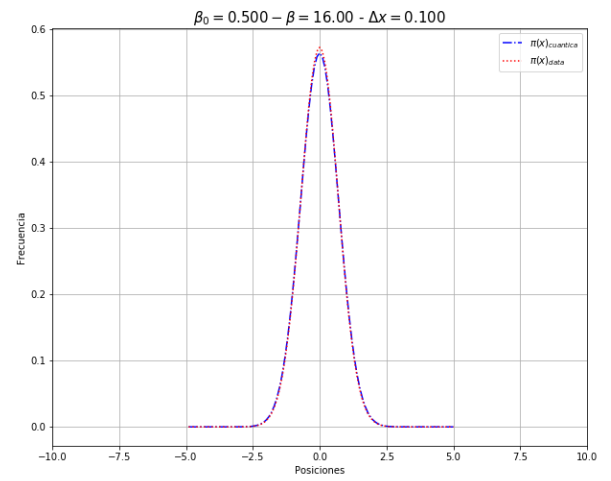
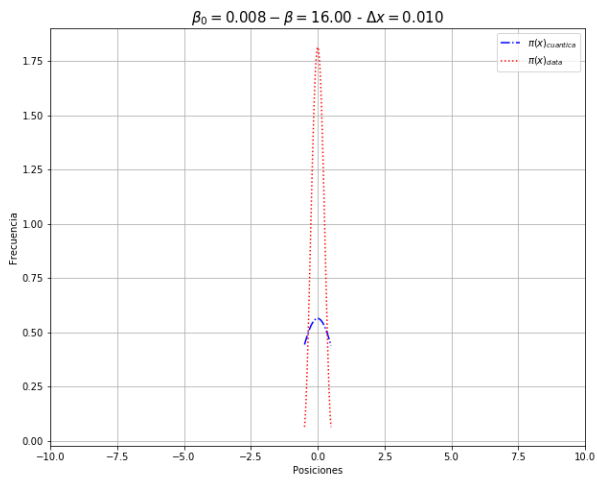
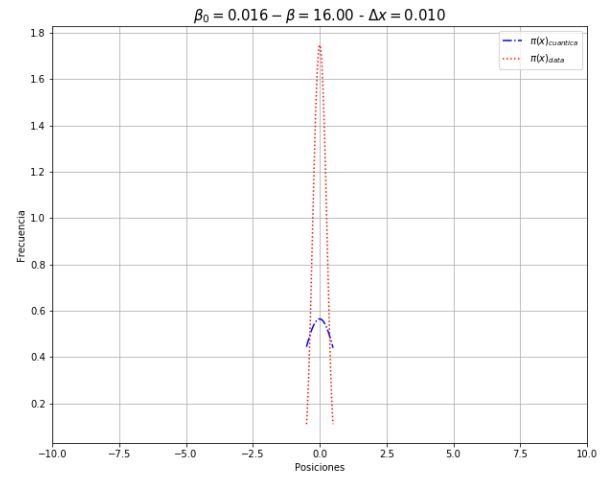
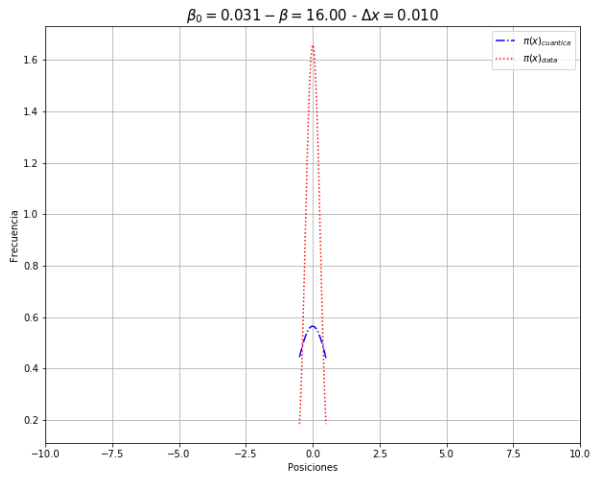
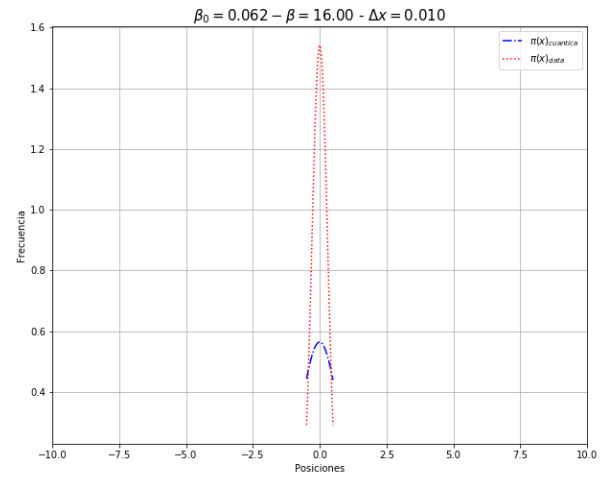
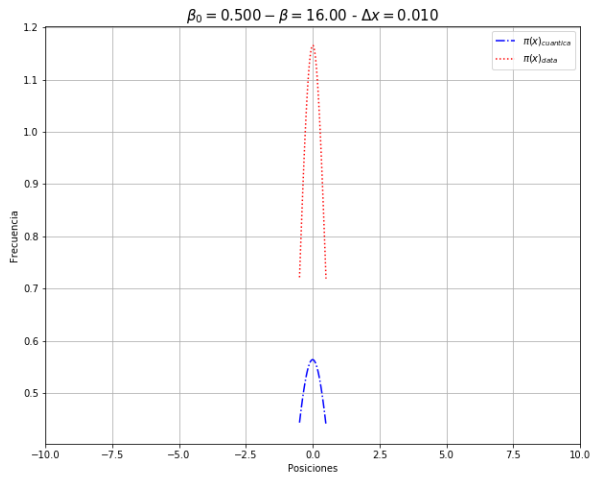
el código se usó $\Delta x = 0,01, 0,1, 0,2, 1$ y con cada uno de esos valores de Δx se varió la temperatura inversa inicial: $\beta_0 = 2^{-1}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}$. Los resultados se muestran en la figura 1. La curva continua negra representa la probabilidad exacta para el caso clásico y se observa que difiere mucho de las demás curvas. Esto tiene sentido pues se está obteniendo la probabilidad para $\beta = 4$ y sólo se espera que la probabilidad cuántica coincida con la clásica en el límite de altas temperaturas ($\beta \rightarrow 0$). Se puede observar que el valor de Δx afecta considerablemente la estimación de la densidad de probabilidad, y se concluye que los mejores resultados se obtienen con $\Delta x = 0,1$. Para la variación en β_0 como nuestro esquema hace una estimación implícita de $\rho(x, x, \beta)$, la matriz densidad sólo se evalúa una vez en el β_0 y lo que se hace es un conteo mediante la variable *beta tmp* para controlar y matener registro de en qué iteración o evaluación de la matriz densidad va el esquema, pero realmente nunca se evalúa en ese *beta tmp* actualizado, sino que se hace una asignación a “mano”. De esta forma la estimación varía al cambiar β_0 . De la forma de la propiedad de convolución se puede concluir que la mejor estimación será para β_0 cercano a la mitad del valor de β para el que se quiere obtener la matriz densidad.

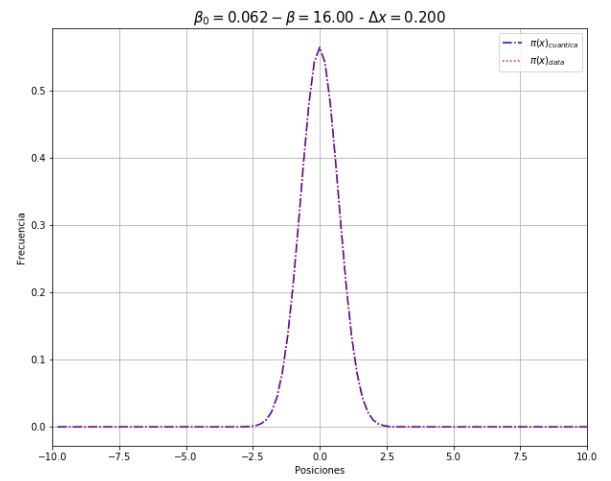
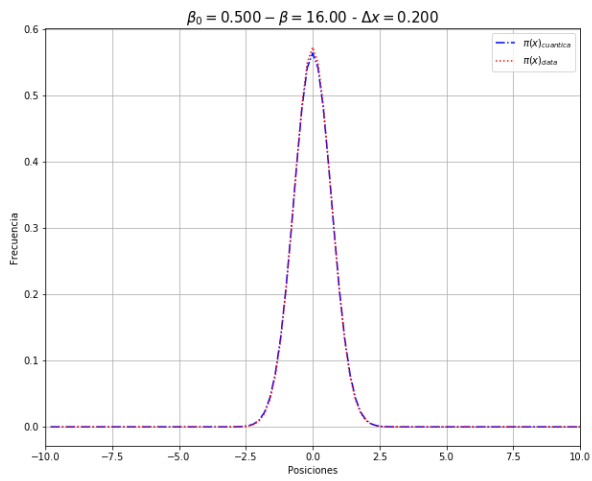
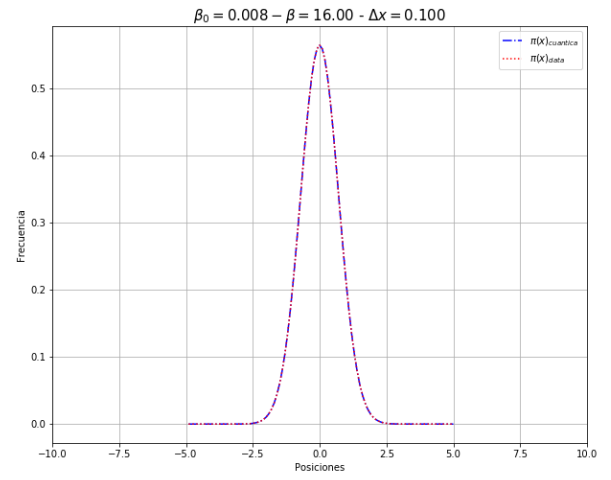
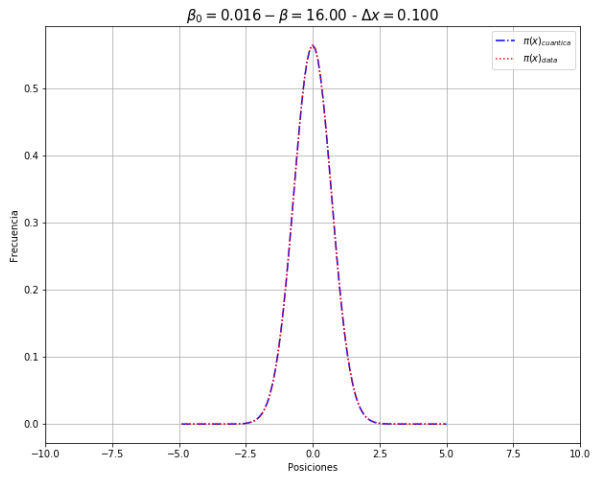
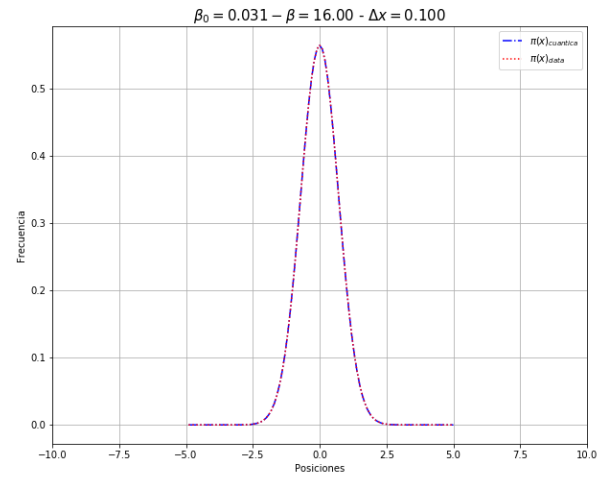
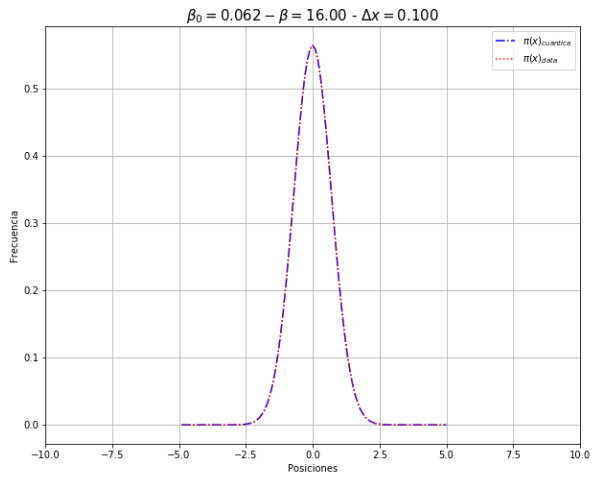
3.2. Integral de Camino usando un muestreo de Monte Carlo

En esta sección usaremos una aproximación de integrales de camino para encontrar la probabilidad de que la partícula se encuentre en una posición dada (oscilando alrededor de dicha posición). Para esto debemos entender que, la propiedad de convolución de la matriz densidad permite, como ya se mencionó, hacer el cálculo de $\rho(x, x', \beta)$ a partir de matrices a $\rho(x, x', \beta/2)$. Es decir que se puede seguir subdividiendo y obtener la matriz $\rho(x, x', \beta/2)$ a partir de $\rho(x, x', \beta/4)$, y así sucesivamente. En general, pasando de $x, x', \dots \rightarrow x_0, x_1, \dots$, se tiene

$$\mathcal{Z} = \int \dots \int dx_1 dx_{N-1} \times \rho\left(x_0, x_1, \frac{\beta}{N}\right) \dots \rho\left(x_{N-1}, x_0, \frac{\beta}{N}\right) \quad (40)$$

Con $\Delta\tau = \beta/N$. Como se mencionó en la introducción, las x_k denotan una configuración o camino del sistema. Cada configuración depende de N variables dinámicas x_i que dan la posición de la partícula en cada nodo temporal τ_k . El objetivo con usar un método de Monte Carlo es evaluar la integral en 40 numéricamente. Lo que hacemos es aproximar la integral con una suma, usando un número finito de términos. El algoritmo de Metropolis está basado en la idea de muestreo por referencia, es decir, los puntos x seleccionados no son completamente aleatorios, sino que están distribuidos con mayor densidad en las regiones del espacio de fase en donde la contribución a la integral es mayor.





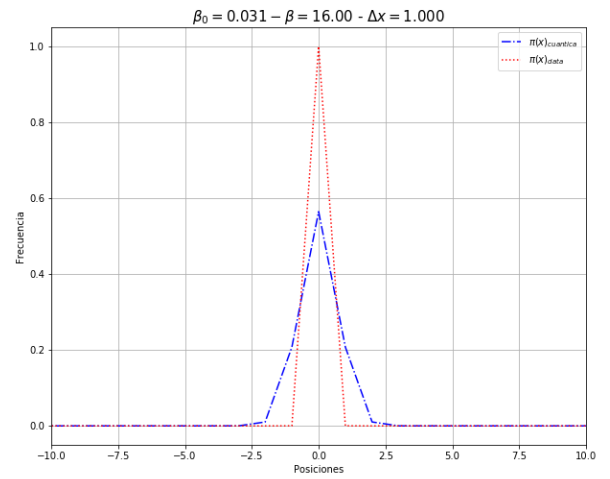
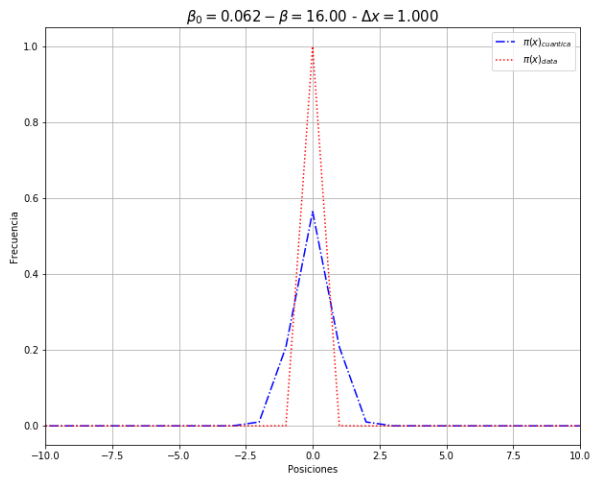
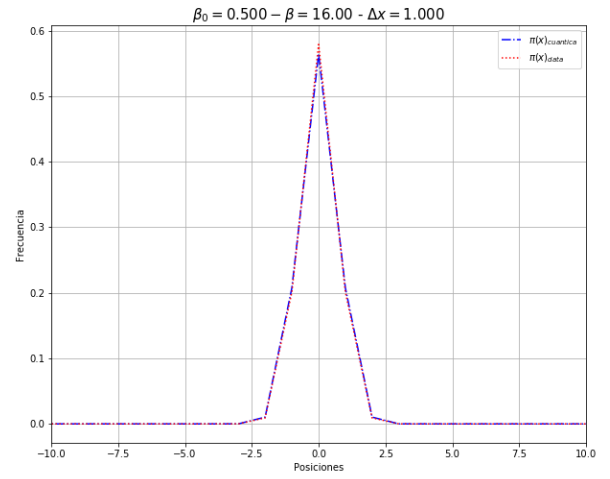
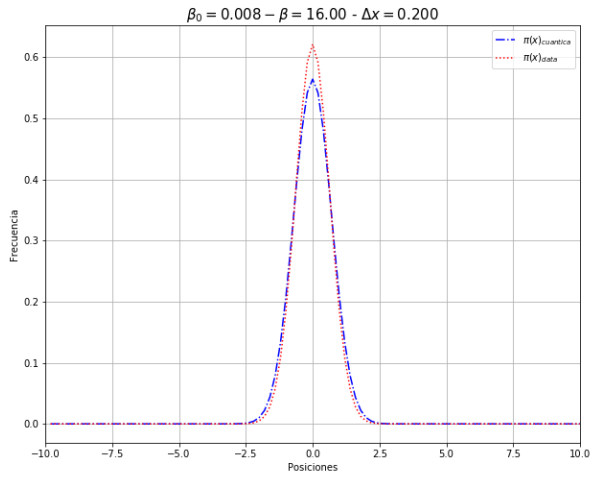
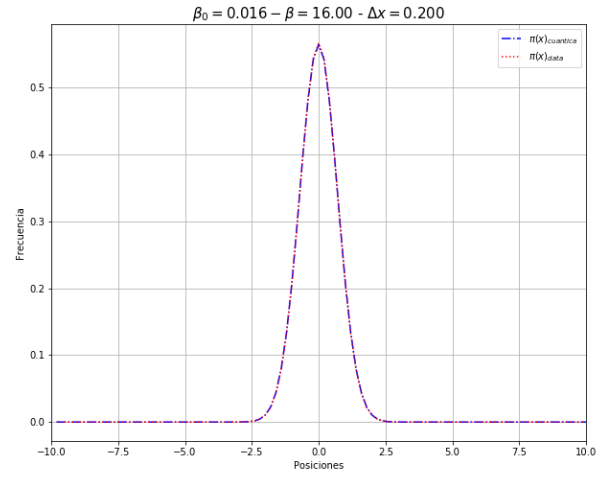
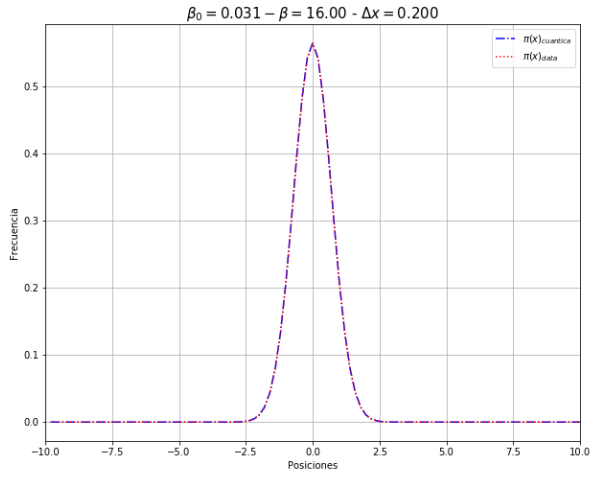


Figura 1: Distribución de probabilidad estimada usando el método de matrix squaring comparada con las curvas teóricas, clásica ($\Pi(x)_{clasica}$) y cuántica ($\Pi(x)_{cuantica}$), para $\beta_f = 4,0$ con distintos valores de β_0 y de Δx .

De la expansión de Trotter se sabe que las probabilidades de los caminos o configuraciones x_k y x'_k son

$$\Pi(x_k) = \rho(x_{k-1}, x_k, \Delta\tau) \rho(x_k, x_{k+1}, \Delta\tau) = \left[e^{-\beta \frac{V(x_{k-1})}{2}} \hat{\rho}^{free}(x_{k-1}, x_k, \Delta\tau) e^{-\beta \frac{V(x_k)}{2}} \right] \quad (41)$$

$$\left[e^{-\beta \frac{V(x_k)}{2}} \hat{\rho}^{free}(x_k, x_{k+1}, \Delta\tau) e^{-\beta \frac{V(x_{k+1})}{2}} \right] \quad (42)$$

$$\Pi(x'_k) = \rho(x_{k-1}, x'_k, \Delta\tau) \rho(x'_k, x_{k+1}, \Delta\tau) = \left[e^{-\beta \frac{V(x_{k-1})}{2}} \hat{\rho}^{free}(x_{k-1}, x'_k, \Delta\tau) e^{-\beta \frac{V(x'_k)}{2}} \right] \quad (43)$$

$$\left[e^{-\beta \frac{V(x'_k)}{2}} \hat{\rho}^{free}(x'_k, x_{k+1}, \Delta\tau) e^{-\beta \frac{V(x_{k+1})}{2}} \right] \quad (44)$$

Luego, para el criterio de aceptación de Metropolis se tiene

$$\frac{\Pi(x'_k)}{\Pi(x_k)} = \frac{\hat{\rho}^{free}(x_{k-1}, x'_k, \Delta\tau) \hat{\rho}^{free}(x'_k, x_{k+1}, \Delta\tau) e^{-\beta V(x'_k)}}{\hat{\rho}^{free}(x_{k-1}, x_k, \Delta\tau) \hat{\rho}^{free}(x_k, x_{k+1}, \Delta\tau) e^{-\beta V(x_k)}} \quad (45)$$

La nueva configuración será $x'_k = x_k + \delta_k$. De esta forma, para elegir x'_k se genera un número aleatorio entre 0 y 1 y la posición es aceptada si se satisface que ese número es menor que $\frac{\Pi(x'_k)}{\Pi(x_k)}$. Para muestrear el movimiento de cada nodo, se seleccionan los nodos k con un número aleatorio entre 0 y el número de nodos totales en los que se ha discretizado el camino: $N - 1$. Las configuraciones o caminos anterior y posterior están dadas por x_{k-1} y x_{k+1} , respectivamente y se estiman a partir de

$$k_{next} = (k + 1) \% N \quad (46)$$

$$k_{prev} = (k - 1) \% N \quad (47)$$

de modo que se garantiza que los caminos sean cerrados, es decir, que se comienza y se termina en el mismo punto, así, cuando el k generado aleatoriamente es igual a $N - 1$, entonces $k_{next} = N \% N = 0$, es decir, el primer y el último nodo coinciden. Así aseguramos tener condiciones de frontera periódicas. Se desarrolló un código en Python para realizar el procedimiento descrito:

```
#Elementos no diagonales de la matriz
#densidad partícula libre

def rho_free(x, xp, beta):
    return (np.exp(-(x-xp)**2/(2.0*beta))\
            /np.sqrt(2.0*np.pi*beta))

#temperatura del baño térmico
beta = 1.0
```

```
#número de nodos en el camino
N = 10
```

```
#discretización del camino
dtau = beta/N
```

```
#desplazamiento máximo de un nodo
delta = 4.0
```

```
#pasos de monte carlo
n_steps = 100000
```

```
#camino inicial
x=[0.0]*N
y=np.arange(0,beta,dtau)
```

```
#potencial teórico
xgrid = np.arange(-3,3,0.1)
potential = 0.5 *(xgrid**2)
```

```
#guardar camino. XS[k][movidas en k]:
#secuencias de desplazamiento de cada nodo
```

```
XS = []
```

```
for j in range(N):
    XS.append([])
```

```
#secuencia de movimientos del nodo 0
xpath0 = []
```

```
x_all = []
```

```
#como son caminos cerrados
#(estoy sumando n_steps posibles caminos para que
#la partícula oscile de una posición a otra)
#al final sí estoy calculando la
#probabilidad de estar en una cierta posición.
```

```
for step in range(n_steps):
    k = random.randint(0,N-1)
```

```
#nodo anterior y posterior:
```

```
knext, kprev = (k+1)%N, (k-1)%N
```

```
#nueva posición: desplazamiento del nodo k
x_new = x[k] + random.uniform(-delta,delta)
```

```
#pi(a)
old_weight = (rho_free(x[kprev], x[k], dtau)*
              rho_free(x[k], x[knext], dtau)*
              np.exp(-0.5*dtau*x[k]**2))
```

```
#pi(b)
```

```

new_weight = (rho_free(x[kprev], x_new, dtau)*
              rho_free((x_new), x[knext], dtau)*
              np.exp(-0.5*dtau*x_new**2))

if random.uniform(0.0,1.0) < new_weight/old_weight:

    #cada vez que se acepta que un nodo se mueva,
    #se está creando un nuevo camino
    x[k] = x_new

    #individuales
    XS[k].append(x[k])

    #camino general
    x_all.append(x[k])

if step%1000==0:
    xpath0.append(x[0])

#Camino de Feynman para algunas iteraciones

if step%10000==0:

    #plot potencial teórico:  $x^2/2$ 
    plt.plot(xgrid, potential)

    #grafica camino
    plt.plot(x, y)

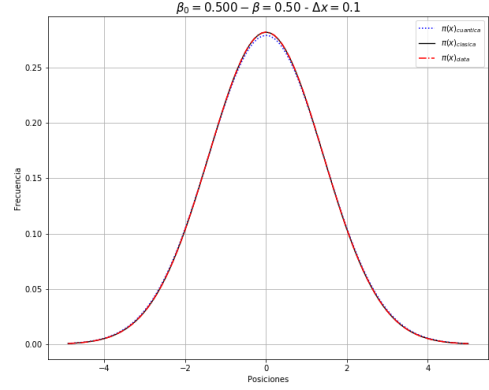
```

Con este código, se obtiene el “paseo aleatorio” de cada nodo k . Se graficaron histogramas normalizados para cada camino x_k y se compararon con el histograma para x_0 . A su vez estos histogramas se comparan con la curva de datos obtenidas con el procedimiento de matrix-squaring de la sección anterior. Los resultados se muestran en las figuras 4

4. Resultados y Discusión

4.1. Oscilador Armónico

Se estimó $\Pi(x)$ para altas temperaturas, es decir, para valores pequeños de β . Se usó una temperatura inversa final dada por $\beta = 0,1$ con parámetros iniciales $\beta_0 = 0,5$ y $\Delta x = 0,1$. Los resultados se muestran en la figura 2. Se observa que las tres curvas coinciden como se esperaba.



(a)

Figura 2: Distribución de probabilidad usando la propiedad de convolución de la matriz densidad para altas temperaturas.

4.1.1. Integral de Camino con Monte Carlo

Con el código para el algoritmo descrito en la sección 3.2, se obtuvieron los histogramas de densidad de probabilidad para cada camino que contribuye a la integral de Feynman, comparados con los datos de la probabilidad obtenida con el procedimiento de matrix-squaring. En la figura 3 se muestra el histograma y el camino de Feynman para $x[0]$ con 10 y 100 iteraciones. Se observa que el muestreo de la distribución de probabilidad sólo es buen para muchas iteraciones (≈ 100000), pues con 10 iteraciones la estadística no es suficiente. Para 100 iteraciones la grafica del camino muestra que los nodos empiezan a muestrear el camino continuo alrededor del cero, lo que coincide con el histograma. En la figura 4 se puede observar que los histogramas coinciden bien con la curva obtenida con el método de matrix-squaring pero la distribución está un poco más aglomerada en $x > 0$ lo que se debe probablemente a la condición inicial del camino.

4.2. Oscilador Anarmónico

Se repitió el procedimiento de matrix-squaring y de la integral de camino con el método de Monte Carlo pero para un potencial anarmónico de la forma

$$V(x) = \frac{x^2}{2} - x^3 + x^4 \quad (48)$$

En este caso, la expansión de Trotter para la matriz densidad es

$$\begin{aligned} \rho(x, x', \beta) &= e^{-\frac{\beta}{2}\left(\frac{x^2}{2} - x^3 + x^4\right)} \hat{\rho}^{free}(x, x', \beta) e^{-\frac{\beta}{2}\left(\frac{x'^2}{2} - x'^3 + x'^4\right)} \\ &= \frac{1}{\sqrt{2\pi\beta}} e^{-\frac{\beta}{2}\frac{x^2 + x'^2}{2}} e^{\frac{\beta}{2}(x^3 + x'^3)} e^{-\frac{\beta}{2}(x^4 + x'^4)} \end{aligned} \quad (49)$$

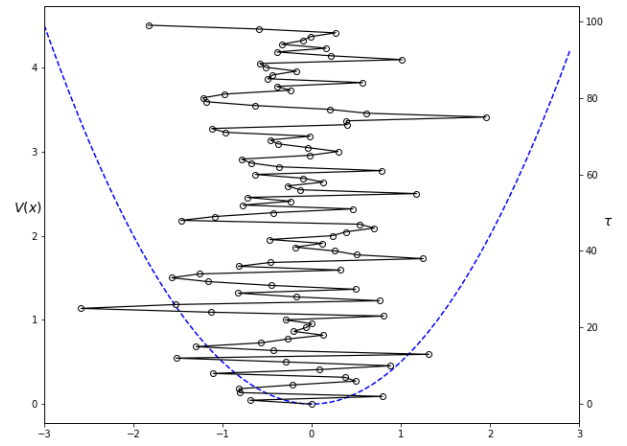
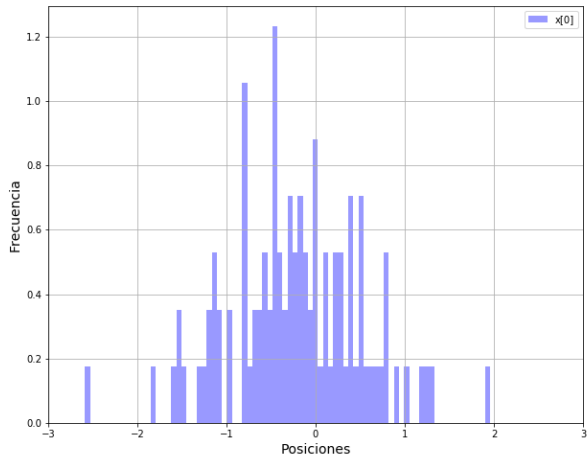
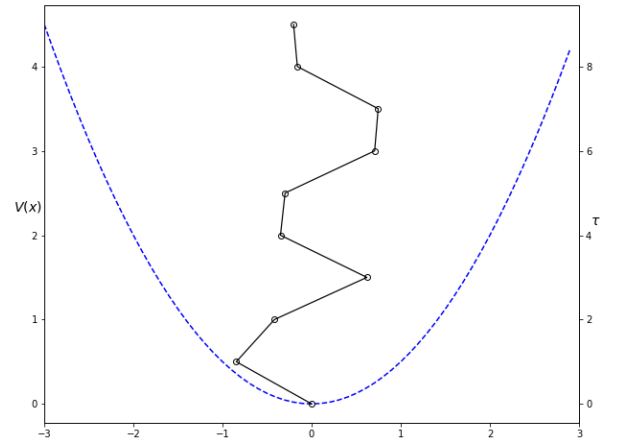
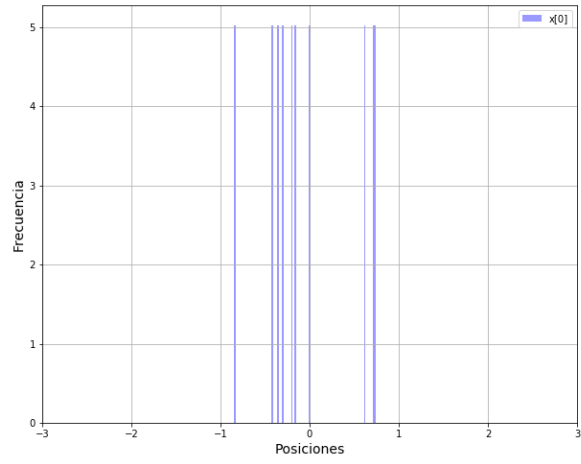


Figura 3: Histograma y camino de Feynman para $x[k=0]$ almacenando cada 10 y 100 iteraciones.

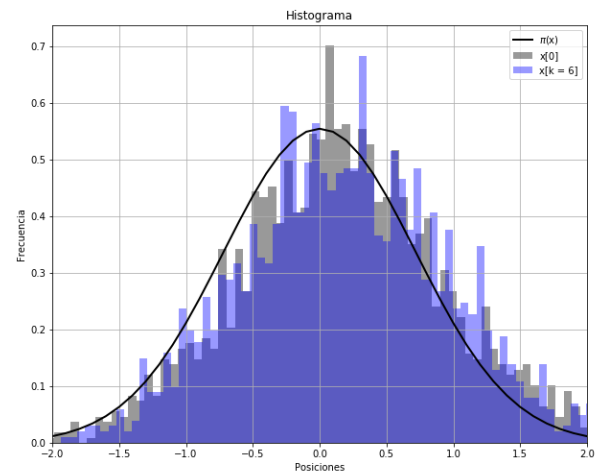
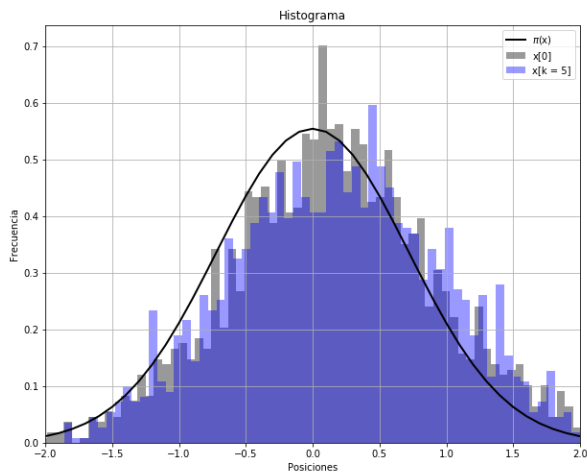
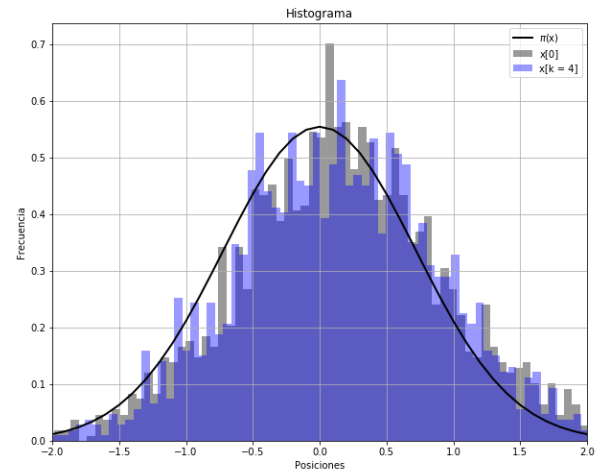
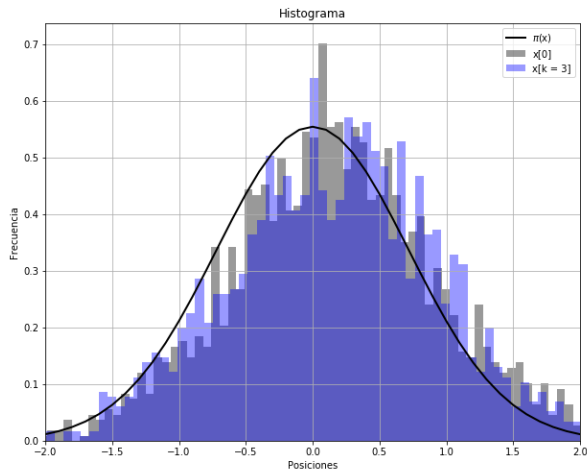
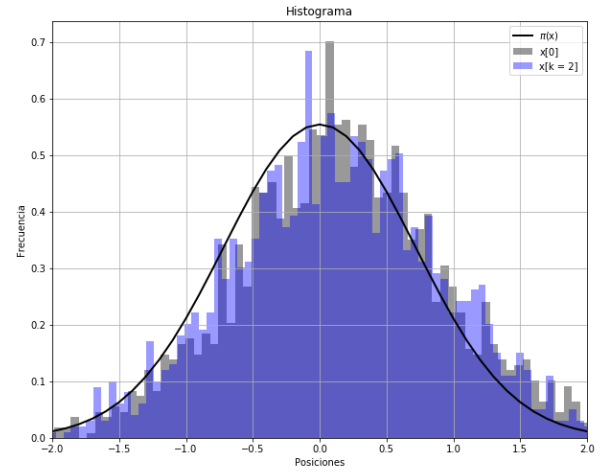
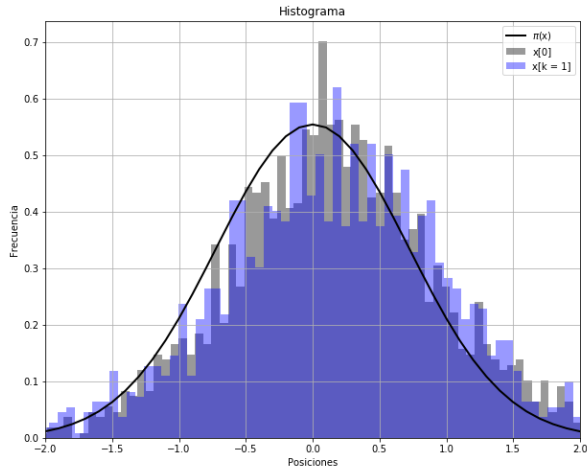


Figura 4: histograma para $x[0]$ comparado con otros caminos $k = 1, 2, 3, 4, 5, 6$. El número de nodos era $N=10$.

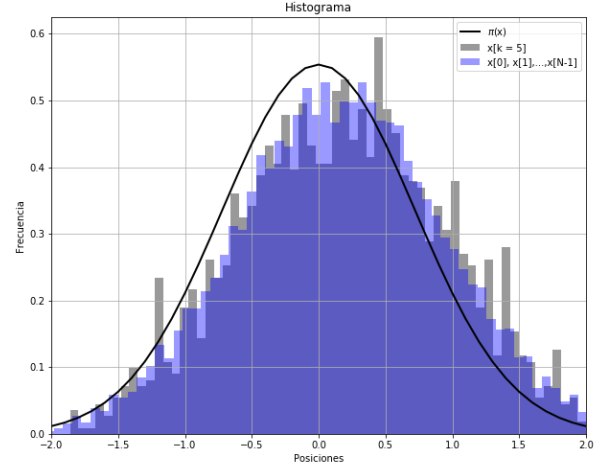
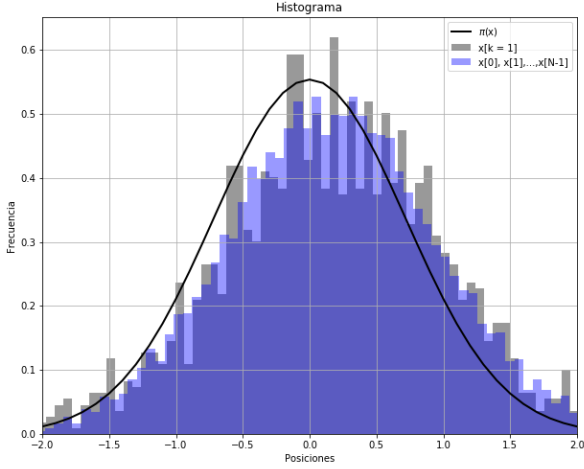
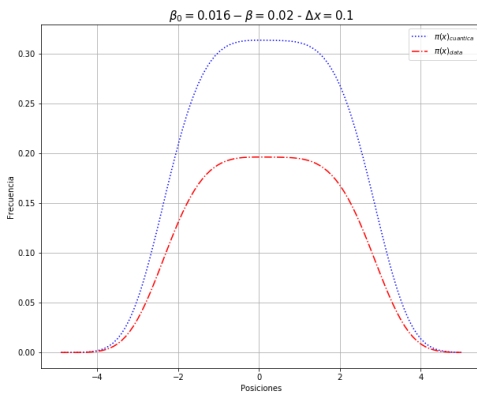


Figura 5: histograma para el camino entero $x[0], x[1], \dots, x[N-1]$ comparado con el del camino $x[1]$ (izquierda) y $x[5]$ (derecha).

Los elementos diagonales de la matriz son

$$\rho(x, x, \beta) = \frac{1}{\sqrt{2\pi\beta}} e^{\frac{-\beta}{2}x^2(1-2x+2x^2)} \quad (50)$$

Con esta nueva forma de la matriz, el procedimiento de matriz-squaring es el mismo que se siguió para el oscilador armónico pero cambiando la ecuación de la matriz densidad. La figura 7 muestra los resultados de este procedimiento. Se observa que la curva teórica no coincide con los datos.



(a)

Figura 7: Método de matrix-squaring para potencial anarmónico.

El algoritmo es el mismo que para el oscilador armónico, sólo que ahora los pesos con que contribuye cada camino

en la integral son diferentes, por lo que para el criterio de aceptación de metropolis, $\frac{\Pi(x'_k)}{\Pi(x_k)}$ es ahora

$$= \frac{\hat{\rho}^{free}(x_{k-1}, x'_k, \Delta\tau) \hat{\rho}^{free}(x'_k, x_{k+1}, \Delta\tau) e^{-\beta\left(\frac{x'^2}{2} - x'^3 + x'^4\right)}}{\hat{\rho}^{free}(x_{k-1}, x_k, \Delta\tau) \hat{\rho}^{free}(x_k, x_{k+1}, \Delta\tau) e^{-\beta\left(\frac{x^2}{2} - x^3 + x^4\right)}} \quad (51)$$

El código para la integral de camino en este caso es entonces

#Elementos no diagonales de la matriz
#densidad partícula libre

```
def rho_free(x, xp, beta):
    return (np.exp(-(x-xp)**2/(2.0*beta)))\
        /np.sqrt(2.0*np.pi*beta))
```

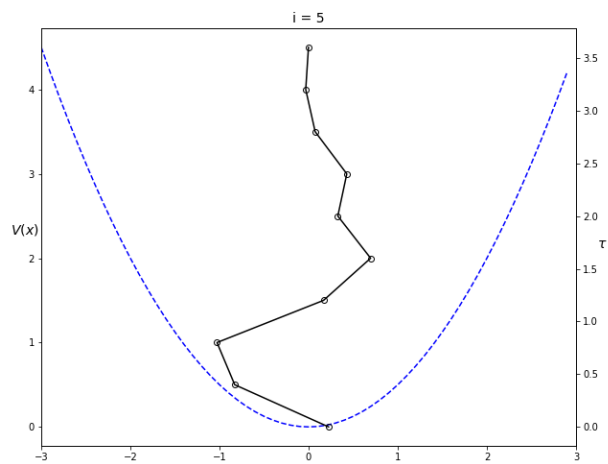
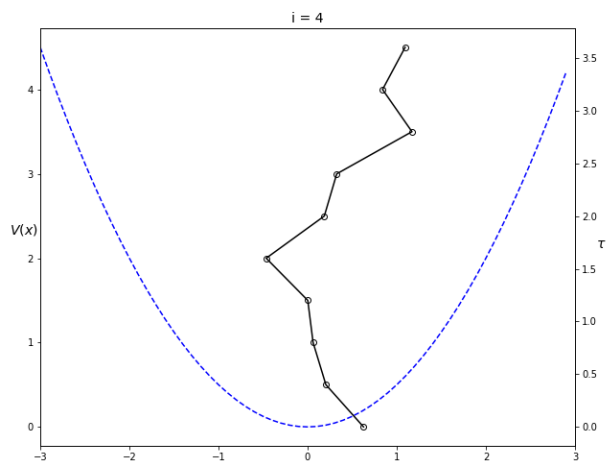
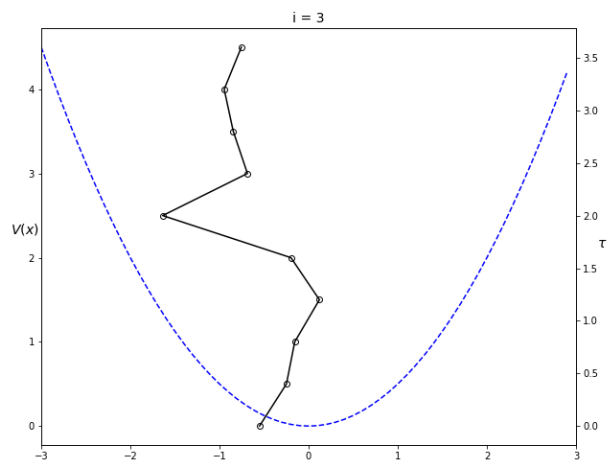
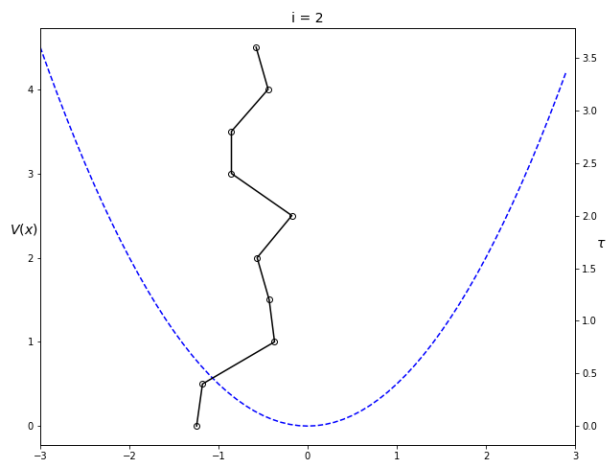
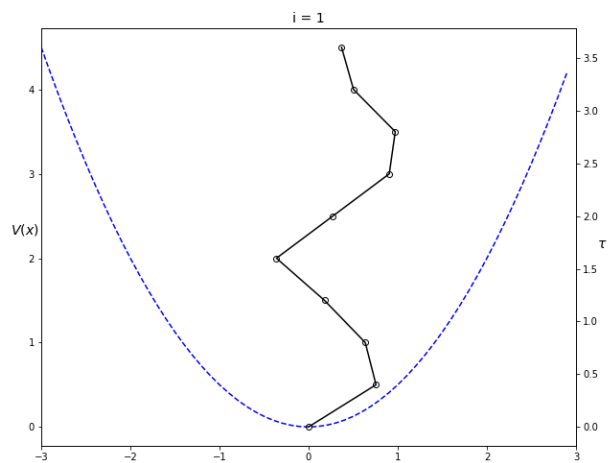
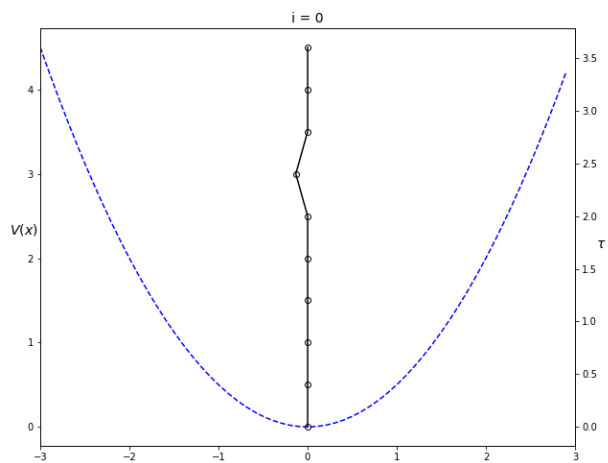
```
beta = 4.0
#número de nodos en el camino
N = 10
```

```
#discretización del camino
dtau = beta/N
```

```
#desplazamiento máximo de un nodo
delta = 1.0
```

```
#pasos de monte carlo
n_steps = 100000
```

```
#camino inicial
x=[0.0]*N
y=np.arange(0,beta,dtau)
```



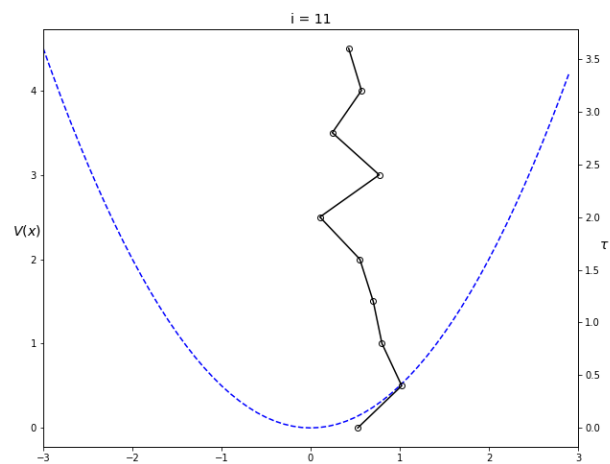
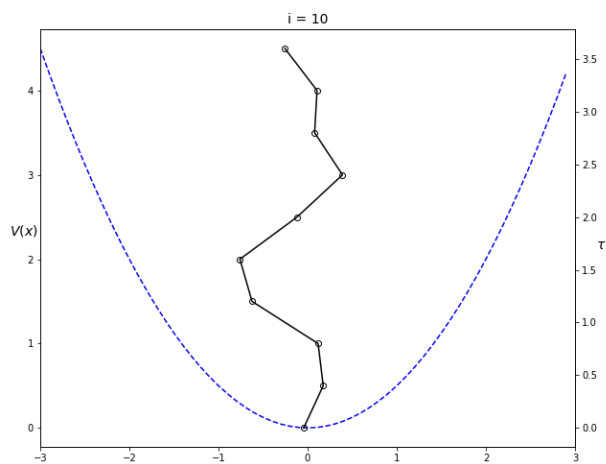
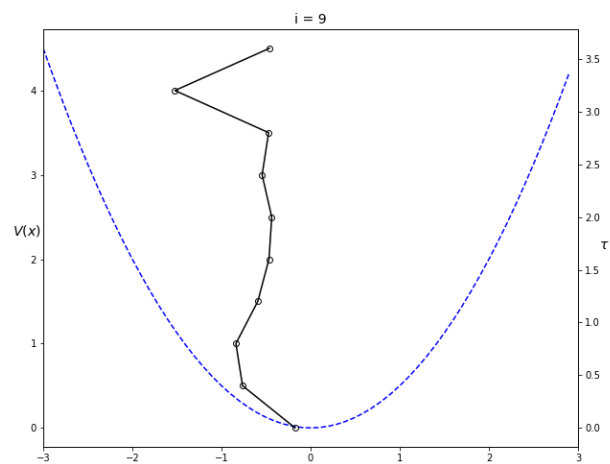
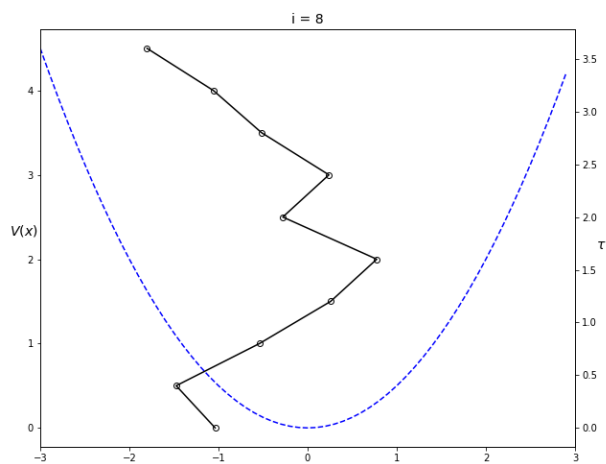
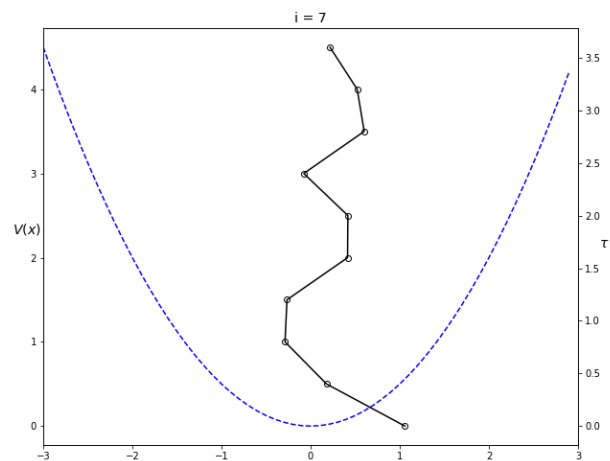
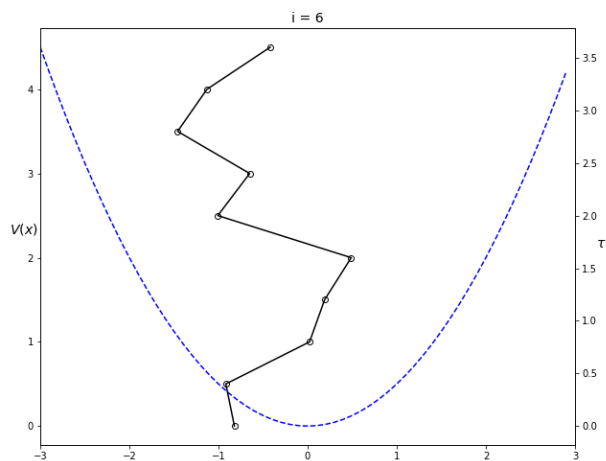


Figura 6: Muestreo de caminos con cadenas de Markov usando el algoritmo Metropolis para un potencial armónico.

```
#potencial teórico
xgrid = np.arange(-3,3,0.1)
potential = \
    (0.5*(xgrid**2) -(xgrid**3) + (xgrid**4))
```

```
#guardar camino. XS[k][movidas en k]:
#secuencias de desplazamiento de cada nodo
```

```
XS = []
```

```
for j in range(N):
    XS.append([])
```

```
#secuencia de movimientos del nodo 0
xpath0 = []
```

```
#
x_all = []
i = 0
```

```
for step in range(n_steps):
```

```
    k = random.randint(0,N-1)
```

```
    knext, kprev = (k+1)%N, (k-1)%N
```

```
    #p(a)
```

```
    old_weight = (rho_free(x[kprev], x[k], dtau)*
        rho_free(x[k], x[knext], dtau)*
        np.exp(-dtau*((0.5*x[k]**2) \
            - x[k]**3 + x[k]**4)))
```

```
    #p(b)
```

```
    new_weight = (rho_free(x[kprev], x_new, dtau)*
        rho_free((x_new), x[knext], dtau)*
        np.exp(-dtau*((0.5*x_new**2) \
            - x_new**3 + x_new**4)))
```

```
    if random.uniform(0.0,1.0) < new_weight/old_weight:
```

```
        x[k] = x_new
```

```
        #individuales
```

```
        XS[k].append(x[k])
```

```
        #camino general
```

```
        x_all.append(x[k])
```

```
    if step%1000==0:
```

```
        xpath0.append(x[0])
```

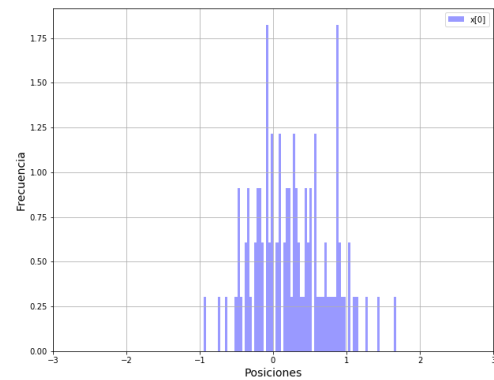
```
    if step%10000==0:
```

```
        #plot potencial teórico:  $x^2/2$ 
        plt.plot(xgrid, potential)
```

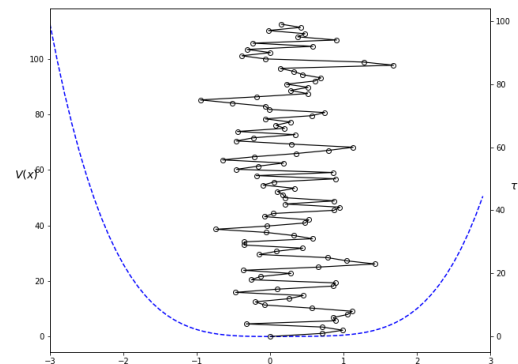
```
#grafica que camino en ésta iteración
plt.plot(x, y)
```

4.3. Integral de Camino

Con el código para el algoritmo descrito en la sección anterior, se obtuvieron los histogramas de densidad de probabilidad para cada camino que contribuye a la integral de la función partición para un potencial anarmónico, comparados con los datos de la probabilidad obtenida con el procedimiento de matrix-squaring. En la figura 8 se muestra el histograma y el camino de Feynman para $x[0]$ con 100 iteraciones. En la figura 9 y 10 se puede observar que los histogramas para cada camino coinciden bien con la curva obtenida con el método de matrix-squaring, mucho mejor que en el caso del potencial armónico. Aquí ambos picos están sobre $x = 0$ y no tienen un exceso de puntos hacia la izquierda favoreciendo $x > 0$.



(a)



(b)

Figura 8: Potencial anarmónico: histograma y camino de Feynman para $x[k = 0]$ almacenando cada 100 iteraciones.

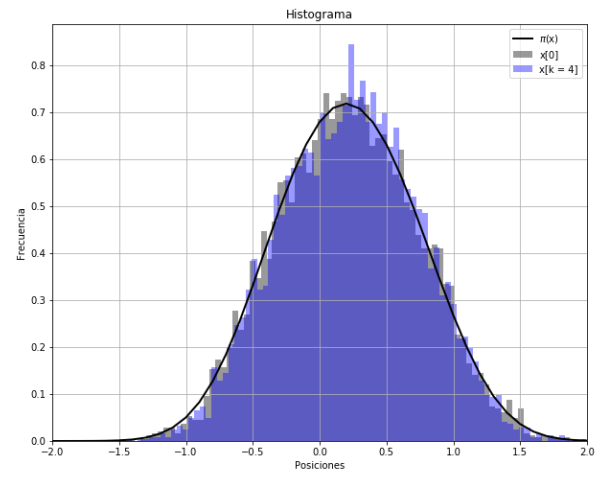
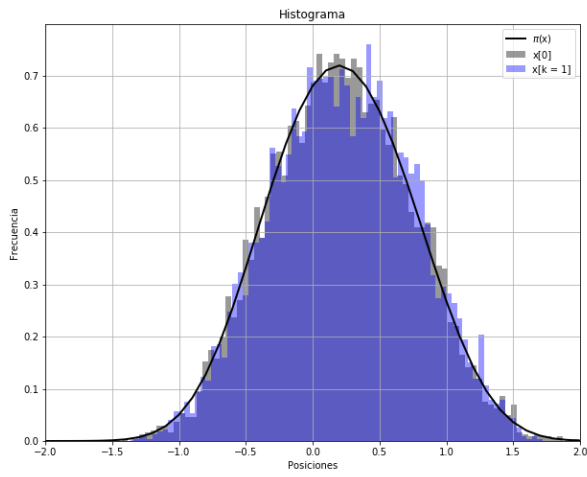


Figura 9: histograma para $x[0]$ comparado con otros caminos $k = 1$ y $k = 4$ para un potencial anarmónico. El número de nodos era $N=10$.

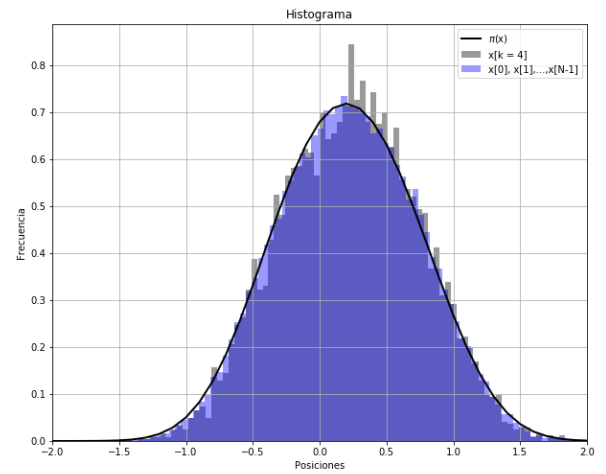
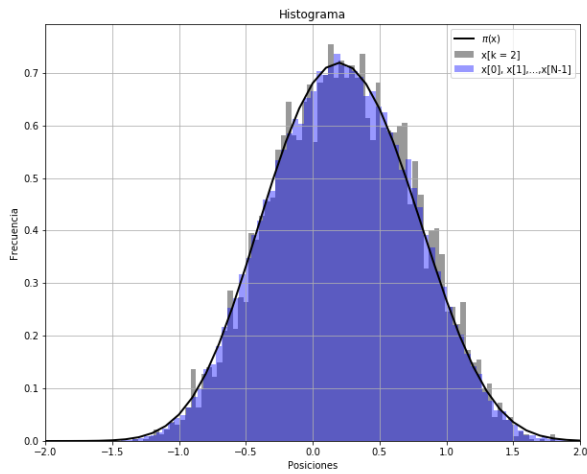
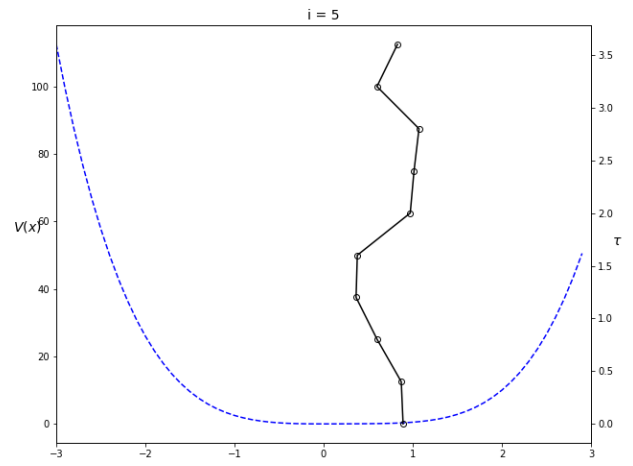
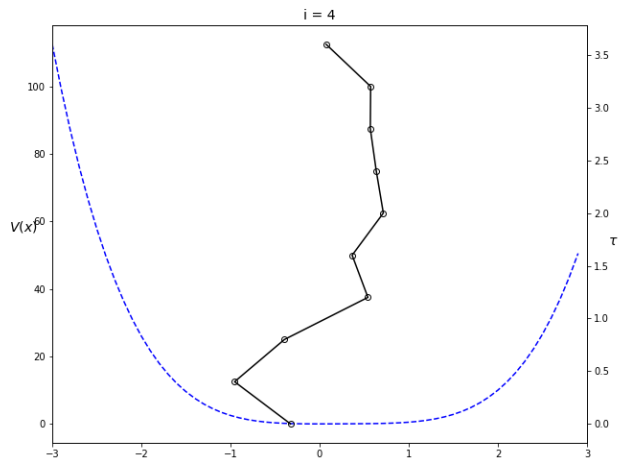
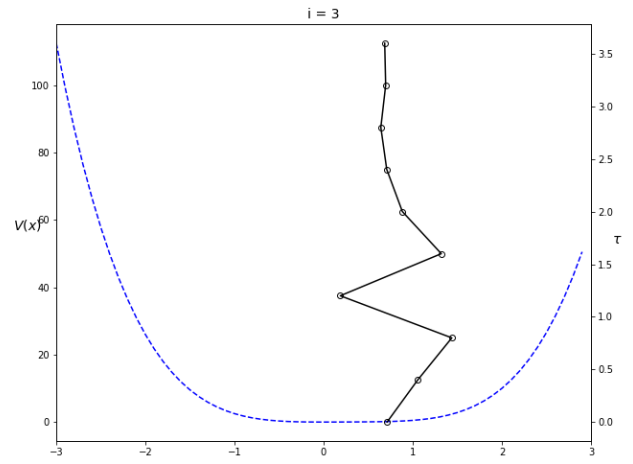
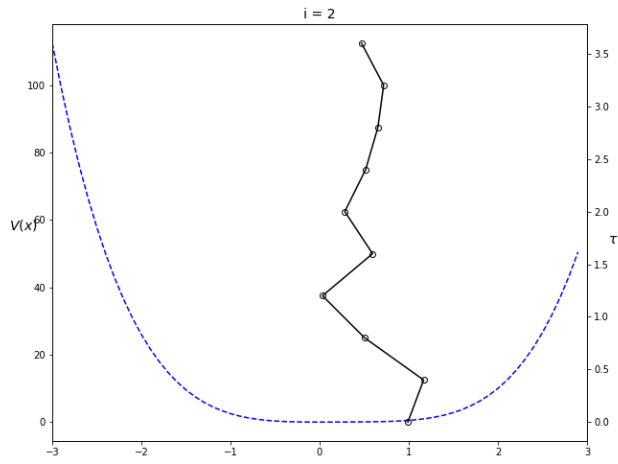
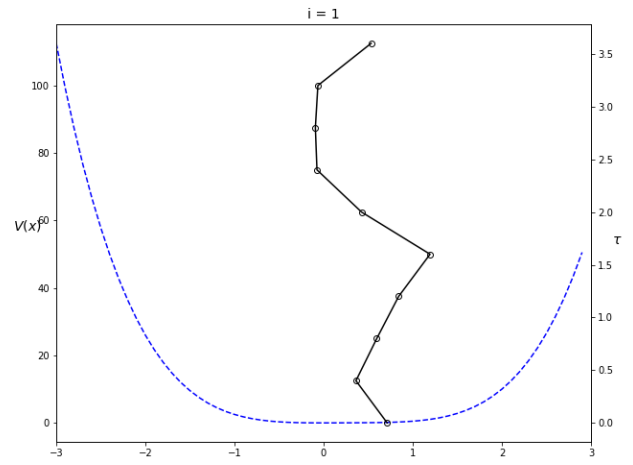
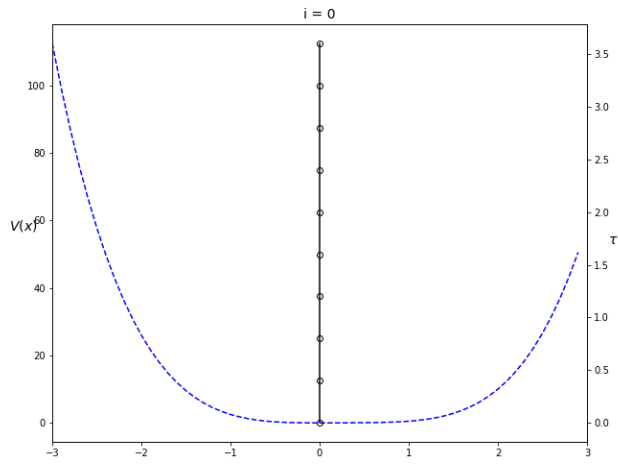


Figura 10: histograma para el camino entero $x[0], x[1], \dots, x[N-1]$ comparado con el del camino $x[2]$ (izquierda) y $x[4]$ (derecha) de una partícula en un potencial anarmónico.



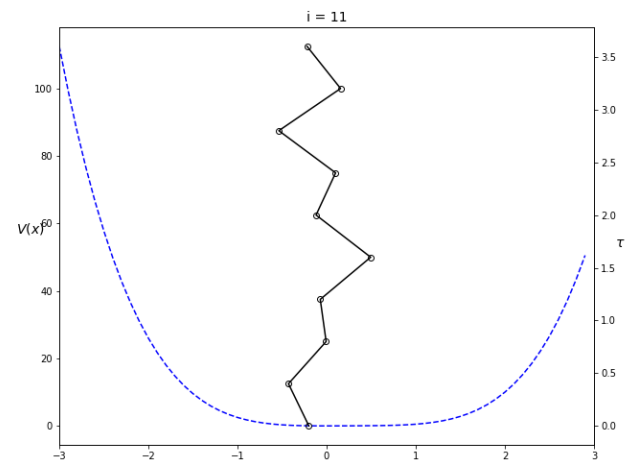
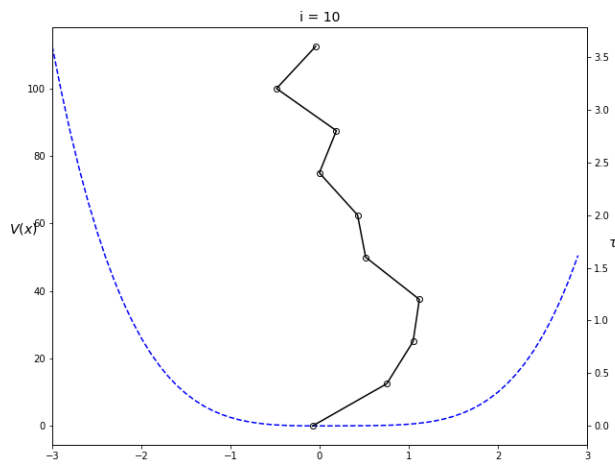
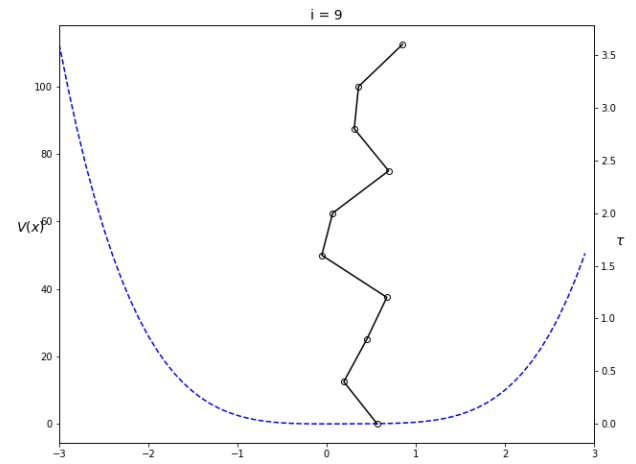
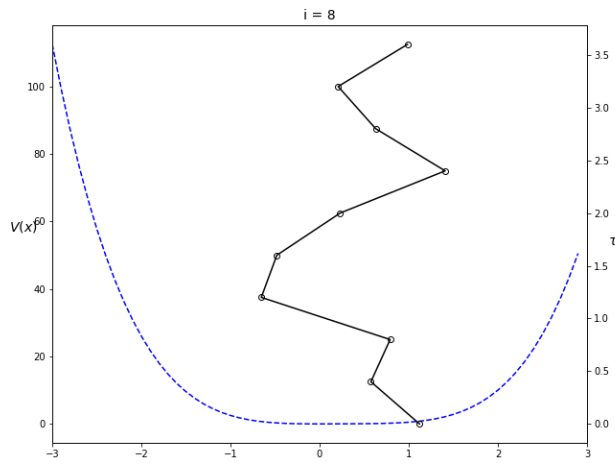
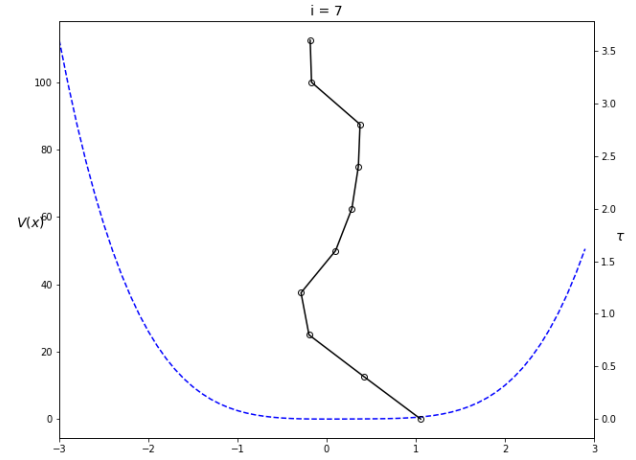
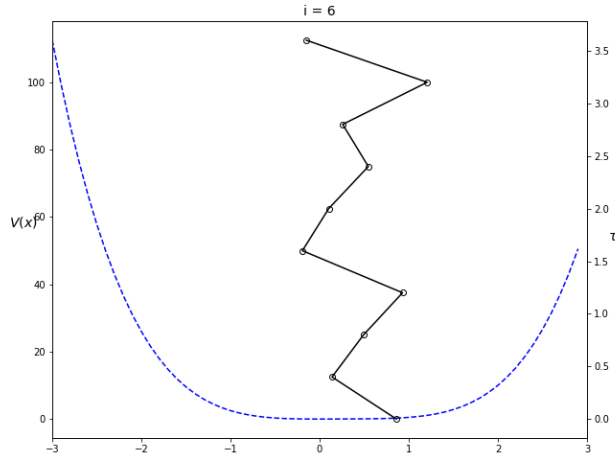


Figura 11: Muestreo de caminos con cadenas de Markov usando el algoritmo Metropolis para un potencial anarmónico.

5. Agradecimientos

Agradezco especialmente a los profesores César Barrero y Johans Restrepo por sus lecciones de física estadística. A Johans Restrepo en particular por los códigos suministrados del algoritmo Metropolis para poder desarrollar este trabajo. A Santiago Quintero por su ayuda sobre herramientas en el lenguaje de programación Python.

Referencias

W. Greiner, L. Neise, and H. Stöcker. *Thermodynamics and statistical mechanics*. Springer Science & Business Media, 2012.

W. Krauth. *Statistical mechanics: algorithms and computations*, volume 13. OUP Oxford, 2006.