Tarea 08: Integración con Montecarlo y uso del Algoritmo de Metropolis

Eva Díaz

November 18, 2015

Curso:	Métodos Numéricos para la Ciencia e Ingeniería
Profesor:	Valentino González
Profesor Auxiliar:	Felipe Pesce

1 Introducción

La tarea con consiste en utilizar el método simple de integración de Montecarlo y el algortimo de Metropolis para resolver dos problemas: el cálculo del centro de masa de un cuerpo y la generación de una muestra aleatoria de números con una distribución dada.

2 Integración por Método de Montecarlo

Se desea estimar la posición del centro de masa de un cuerpo generado por la intersección de un toro con un cilindro. Las ecuaciones (1) y (2) describen estos cuerpos, mientras que en la Figura 1 se ilustra su intersección.

$$\text{Toro}: \; z^2 + \left(\sqrt{x^2+y^2} - 3\right)^2 \leq 1 \tag{1}$$

$$\text{Cilindro}: \; (x-2)^2 + z^2 \leq 1 \tag{2}$$

Para encontrar el centro de masa de este sólido se utiliza la fórmula conocida (3) para cada una de las coordenadas, donde ρ es la densidad del sólido y está dada por $\rho(x,y,z) = 0.5 * (x^2+y^2+z^2)$.

$$X_{cm} = \frac{\int x\rho dxdydz}{\int \rho dxdydz} \tag{3}$$

Para encontrar el valor de estas integrales se utiliza el método de Monte Carlo, que consiste en tomar un volumen con forma de paralelepípedo lo más pequeño posible que encierre el volumen a integrar. Dentro del volumen mayor se toman

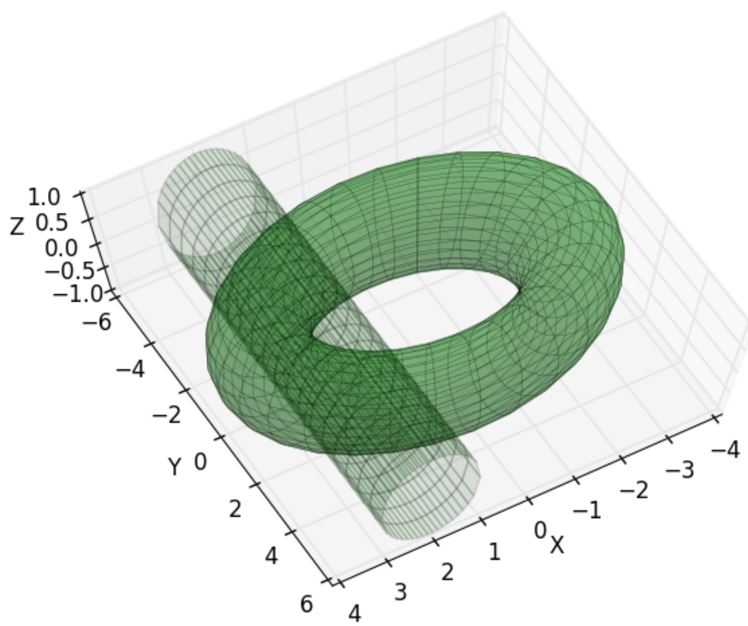

Figure 1: Intersección del toro y el cilindro. Créditos de la imagen a Fernanda Pérez.

puntos aleatorios (obtenidos mediante la función `random.uniform` del módulo `numpy`) y si éstos pertenecen al volumen menor, entonces se agregan a la suma ponderados por la densidad local de masa.

El centro de masa calculado para una muestra de 5000 puntos aleatorios tiene coordenadas:

$$\begin{aligned} x_{cm} &= 2.07 \pm 0.05 \\ y_{cm} &= -0.01 \pm 0.05 \\ z_{cm} &= 0.012 \pm 0.009 \end{aligned} \tag{4}$$

Para una muestra aleatoria de 15000 puntos el centro de masa calculado es:

$$
\begin{aligned}
x_{cm} &= 2.07 \pm 0.02 \\
y_{cm} &= 0.02 \pm 0.03 \\
z_{cm} &= 0 \pm 0.005
\end{aligned} \tag{5}
$$

Para una muestra de 750 puntos las coordenadas del centro de masa son:

$$
\begin{aligned}
x_{cm} &= 2 \pm 0.1 \\
y_{cm} &= -0.1 \pm 0.1 \\
z_{cm} &= 0 \pm 0.02
\end{aligned} \tag{6}
$$

Se puede notar que la cantidad de puntos aleatorios influye principalmente en el error en torno a las coordenadas del centro de masa. Entre menos puntos se evalúen mayor es el rango en el que se ubica la coordenada, lo cual es un resultado evidente y esperado. En tanto, las coordenadas del centro de masa están en concordancia con lo que se puede observar en la Figura 1: el volumen a integrar tiene simetrías, por lo que es de esperar que su centro de masa se ubique en el centro de éstas. La densidad varía aumentando con el radio de una esfera centrada en el origen, lo cual aporta al hecho de que el centro de masa se ubique en una posición en la que hay simetrías.

3 Uso del Algoritmo de Metropolis

Se desea generar una muestra aleatoria de números que sigan una distribución dada por la siguiente ecuación:

$$
W(x) = 3.5 \times \exp\left(\frac{-(x-3)^2}{3}\right) + 2 \times \exp\left(\frac{-(x+1.5)^2}{0.5}\right) \tag{7}
$$

Esta distribución no está normalizada, sin embargo, esto no representa un problema para su uso en el algoritmo de Monte Carlo. Para efectos de graficar, al normalizarla a la unidad se obtiene que su valor es:

$$
W_{norm}(x) = \frac{W(x)}{\sqrt{\pi}(3.5\sqrt{3} + 2\sqrt{0.5})} \tag{8}
$$

Para generar la muestra se utiliza el algoritmo de Metropolis con una distribución $x_p = x_n + \delta r$, donde δ es un parámetro tal que la cantidad de puntos aceptados por el algoritmo sea aproximadamente el 50% de los puntos totales de la muestra, que en este caso serán 10 millones. Por ensayo y error se determinó que el valor $\delta = 4.5$ arroja 5065292 puntos aceptados, por lo cual nos quedamos con este valor.

El algoritmo de Metropolis compara el cuociente entre $W(x_p)$ y $W(x_n)$ (donde x_n es el enésimo punto de la muestra) con un parámetro aleatorio h con distribución uniforme entre -1 y 1 (nuevamente se utiliza la función `random.uniform`

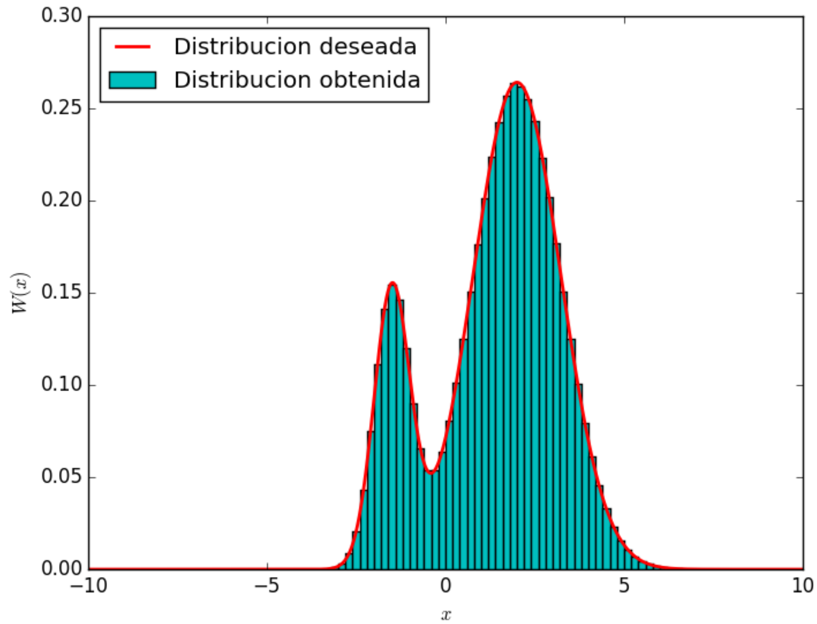

Figure 2: Distribución obtenida y requerida de una muestra de puntos.

del módulo `numpy`). Si h es mayor a este cuociente, entonces se toma el valor x_p para el siguiente punto, y si no, se toma el valor de x_n. Se dice que para el primer caso el punto ha sido aceptado.

Finalmente, se grafican los puntos de la muestra aleatoria evaluados en la distribución $W(x)$ junto con la curva teórica que debiesen decribir estos valores. Se observa en la Figura 2 que la muestra aleatoria obtenida se ajusta muy bien a la curva teórica, lo cual indica que el método ha sido bien implementado.

4 Conclusiones

Para el primer problema se obtuvo los valores intuitivamente esperados de acuerdo a la simetría del cuerpo ($x = 2$, $y = 0$, $z = 0$) para varios tamaos de la muestra, con errores de estimación pequeños (alrededor de 10^{-2}). Esto indica que la integración a través del método de Monte Carlo fue realizada exitosamente.

En el segundo problema se obtuvo satisfactoriamente una muestra que se ajusta muy bien a la distribución requerida para un valor de $\delta = 4.5$. Lo observado en la Figura 2 permite comprobar que la muestra obtenida se ajusta muy bien a la curva teórica, lo cual indica que el algoritmo fue aplicado exitosamente.