Métodos Numéricos para la Ciencia e Ingeniería:
Informe Tarea8

Martín Panza
RUT 18.954.534-7

17 de Noviembre, 2015

1. Pregunta 1

1.1 Introducción

El objetivo fue estimar la posición del centro de masa, mediante un algoritmo de integración de MonteCarlo, un volumen obtenido al intersectar un toro y un cilindro de ecuaciones:

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

Cuya densidad seguía la siguiente relación:

$$\rho(x, y, z) = 0.5 * (x^2 + y^2 + z^2)$$

1.2 Procedimiento

Siguiendo la sugerencia dada en el enunciado, el procedimiento se basó en lo explicado en la sección "Simple MonteCarlo Integration" del libro "Numerical Recipes in C."

En primer lugar, se debió notar que, debido a la intersección de los volúmenes, los dominios de las variables fueron: x e [1, 3] ; y e [-4, 4]; z e [-1,1]. Luego, se definió el número de pasos a utilizar en N=10000000. También se utilizó la semilla 7000. Esto debido a que para cada N se otorgó un valor aleatorio para cada variable dentro de su dominio. Por último, se realizó la integración dentro de un cilindro de radio 1 y altura 8.

A continuación se procedió a iterar N veces de manera que para cada iteración se comprobara si los tres valores asignados a las variables cumplían con la condición de estar dentro de la figura objetivo. Esta condición se obtuvo de restar ambas inecuaciones. Esto fue implementado en la siguiente sentencia 'if':

```
if ((math.sqrt(x ** 2 + y ** 2) - 3.0) ** 2 ) - ((x - 2) ** 2) < 0.0:
```

En caso de que se cumpliera, se añadía a una suma (para cada valor), la multiplicación entre el valor momentáneo en la iteración y la densidad. También guardaba en una suma un estimador de error.

Una vez terminadas las iteraciones se recogían las posiciones de centro de masa al multiplicar por el volumen y dividir por N. También se obtuvo el error asociado a cada variable.

1.3 Resultados

Posición x = 92.7252788611
Posición y = 0.197027480135

Posición z = -0.0531583475593

Error x = 0.141274415565
Error y = 0.168595227567
Error z = 0.0440045251781

1.4 Conclusiones

El resultado obtenido para la posición x es inverosímil pues escapa de sus propios límites. Por otro lado, los valores para z e y tienen sentido pues intuitivamente debiesen aproximarse a 0 al existir simetría. Para la amplia muestra utilizada se obtuvieron errores considerables pero razonables.

2. Pregunta 2

2.1 Introducción

El objetivo de esta actividad fue obtener una muestra aleatoria de números según la distribución:

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

2.2 Procedimiento

El procedimiento consistió en el uso del algoritmo de Metrópolis. En primer lugar se utilizó la muestra pedida de N=10000000 puntos de muestra. Ya que la función no estaba normalizada, se procedió a normalizarla calculando su integral impropia entre $-\infty$ e ∞. Este resultado arrojó 13.25155870806912. Luego, dividiendo W por este valor se obtenía W normalizada.

Habiendo hecho esto, se graficó esta distribución para analizarla. En efecto, se notó que para x entre -5 y 10 se obtenía la información más importante. Por lo que se trabajó en este intervalo.

Ya que el algoritmo de metrópolis trabaja con números aleatorios se determinó una semilla (50) para obtener valores de una distribución uniforme entre 0 y 1. Pero además, fue pedido utilizar xp(x de prueba) de la forma xp = xn + δ * r; con r un valor aleatorio obtenido de otra distribución uniforme entre -1 y 1. Por lo tanto, se creó otro arreglo de valores aleatorios con otra semilla además (51).

δ se debía elegir de forma que se aceptara un 50% de los valores. El valor elegido fue 4.

A continuación se iteró según el algoritmo de Metrópolis, en el que se evalúa xp y xn en W tal que si su división es mayor que el número aleatorio, se guarda la prueba en xn+1. Es decir:

```
for i in range(0, N-1):
    if  W(x_p)/W(x[i])> r[i]:
        x[i+1] = x_p
        contador+=1
    else:
        x[i+1] = x[i]
```

Además puede observarse en el fragmento del código que se hizo uso de un contador que nos indicara la cantidad de pruebas aceptadas. Esto para evaluar nuestro δ elegido.

Finalmente se graficó un histograma con 50 bins.

2.3 Resultados

Utilizando $\delta=4$ se obtuvieron los siguientes porcentajes de pruebas aceptadas al variar las semillas

Semilla1	Semilla2	Porcentaje
50	51	49.96441%
54	55	49.96019%
540	5	49.99657%

Tabla1: Porcentaje de pruebas aceptadas para $\delta=4$

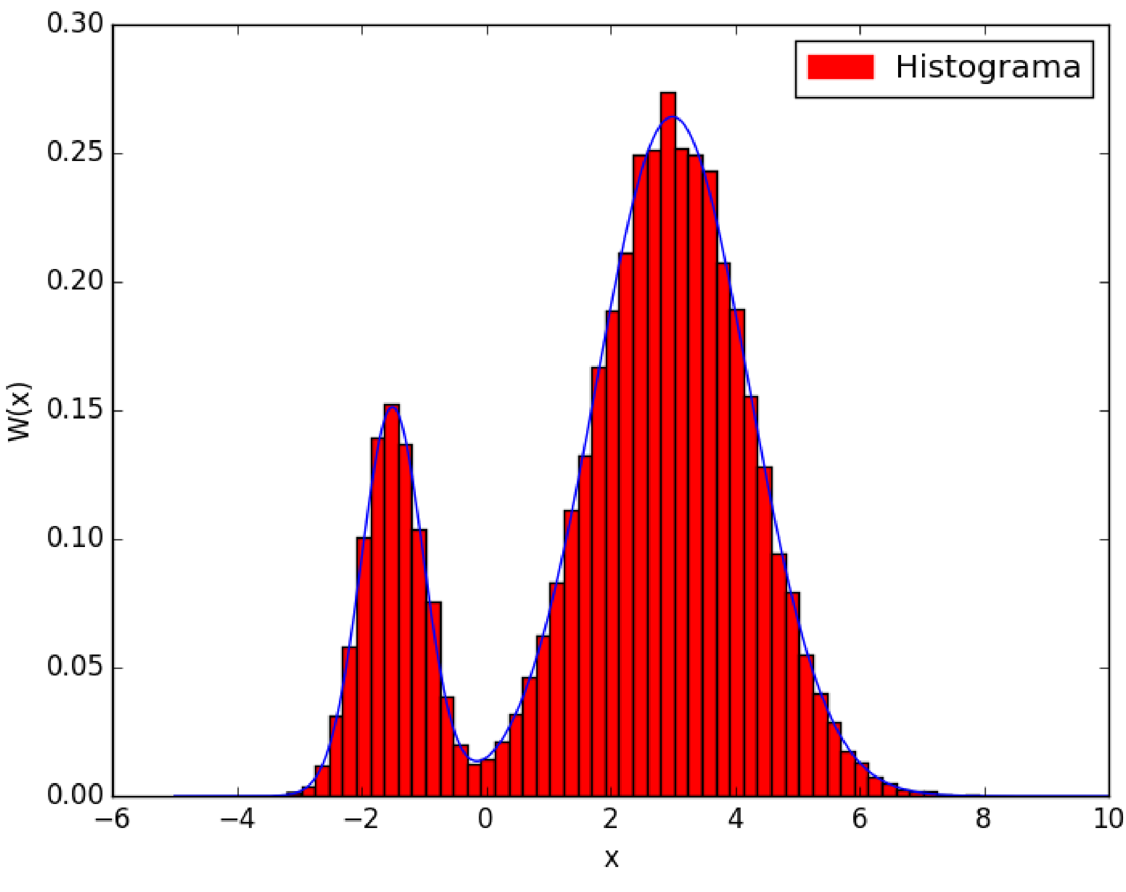

Figura1: Gráfico de W(x) en azul con su histograma en rojo.

2.4 Conclusiones

Tomando $\delta=4$ se alcanza una muy buena aproximación al valor que buscábamos de 50%, aún al cambiar las semillas.

De la figura1, podemos observar que el histograma de la distribución luego de utilizar el algoritmo de metrópolis nos da una buena aproximación de la distribución original.

3. Parte Extra

3.1 Introducción

En esta parte, se buscó determinar la incertidumbre asociada a cada bin del histograma, graficando el histograma incluyendo las barras de error respectivas

3.2 Procedimiento

Básicamente, lo que se realizó en este caso fue el mismo proceso que en la parte 2. Sin embargo, en este caso se realizó aquel proceso n=100 veces distintas para diferentes semillas. Pero el tiempo que tardaba era demasiado grande, por lo que el número de muestras N se debió disminuir a 100000.

De esto se obtuvo una lista con n histogramas diferentes. Para calcular, se analizó la desviación estándar para cada bin correspondiente a los histogramas. Así, se graficó el histograma con sus barras de error utilizando la función 'plt.bar'.

3.3 Resultados

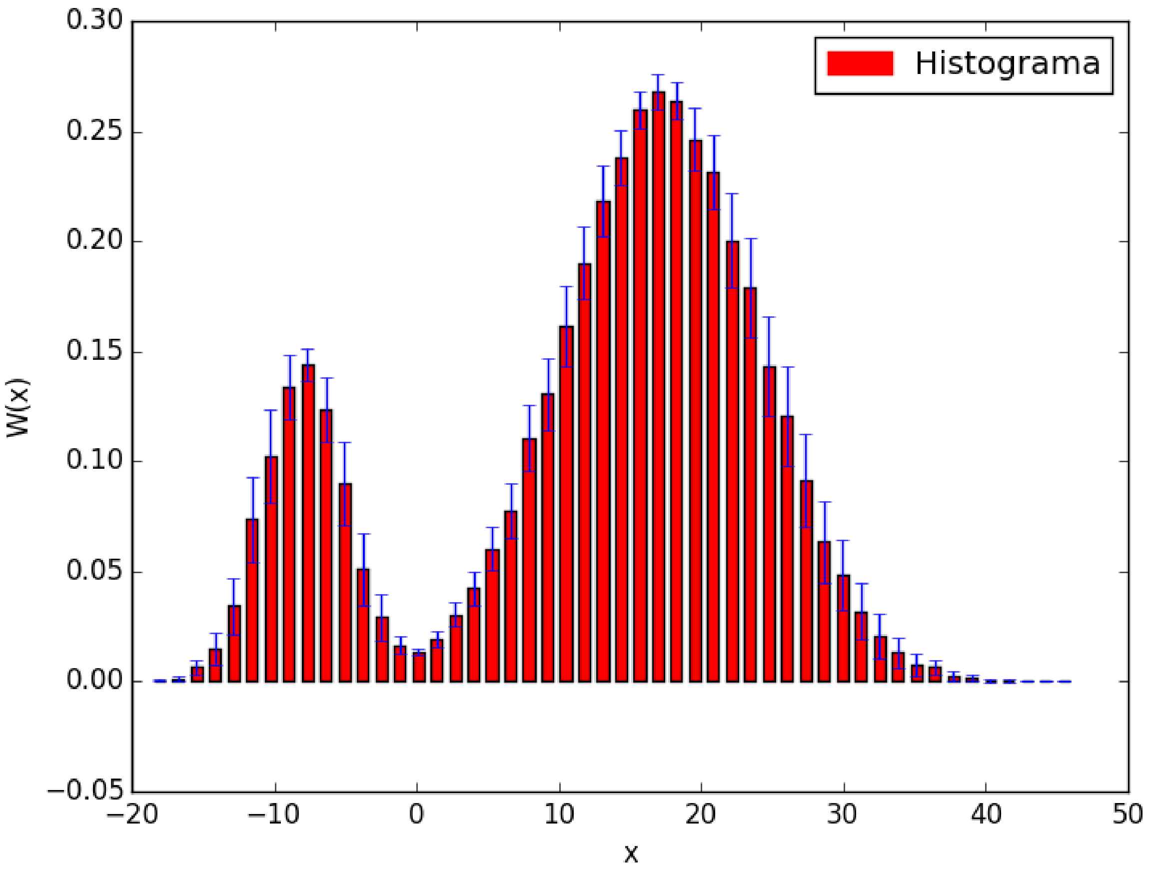

Figura2: Histograma con barras de error asociadas.

Incertidumbre asociada a cada bin:

[0.00039707 0.0010648 0.00315931 0.00713768 0.01298811 0.01946216
0.02130477 0.01459331 0.00749656 0.01452302 0.0187982 0.01629175
0.01051541 0.00406149 0.00126235 0.00359846 0.00555422 0.00767991

0.00979512 0.01248966 0.01496178 0.01653069 0.01837933 0.01644323
0.01606033 0.01254599 0.00836271 0.00807573 0.0084124 0.01436193
0.01650116 0.02154419 0.0226288 0.02286845 0.02260366 0.02129596
0.01849977 0.01586343 0.01290554 0.0099701 0.0068648 0.0051121
0.00348425 0.00227655 0.00147317 0.00095722 0.00054943 0.00033774
0.00018472 0.00015577]

Figura 3: Arreglo de error en bins

3.4 Conclusiones

Como puede observarse en la figura 2, los bins con mayor error asociado se encuentran en los sectores de la distribución donde la pendiente es más inclinada. Asimismo, en las puntas o en los valles se observa un menor error. Esto tiene sentido, pues dado que un bin determina una línea recta dado un intervalo, una línea más vertical tendrá una peor aproximación.

Además, del arreglo se desprende que la incertidumbre asociada a un bin fluctúa entre [0.00015577 ,0.02286845]. El promedio en aquel caso resulta 0.01065. Esto parece ser una incertidumbre bastante razonable. Lo más probable es que disminuya para una muestra más amplia, pero a un alto costo en tiempo.