

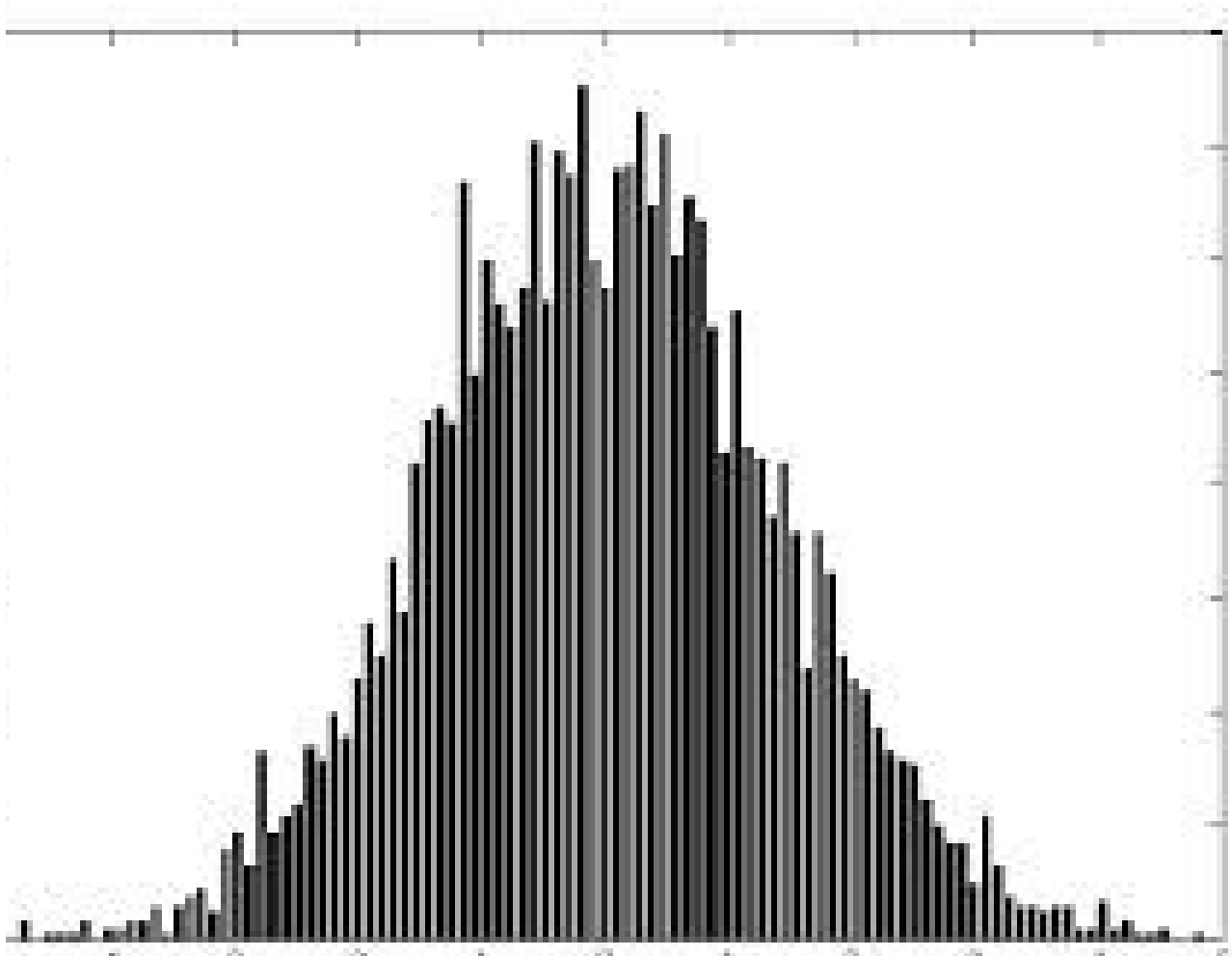
*Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas*

***FI3104-1 MÉTODOS NUMÉRICOS PARA LA CIENCIA E  
INGENIERÍA***

***TAREA N°8***

***“ Utilidades del Método de Monte Carlo y Algoritmo de  
Metrópolis”***

*Benjamín Venegas  
15 de noviembre de 2015  
Santiago, Chile*



## **Resumen**

*En el presente trabajo se buscaba encontrar el centro de masa de un cuerpo sólido y adecuar los valores aleatorios  $x$  de la mejor forma posible a una función de distribución de probabilidad (no normalizada)  $W(x)$  dada, utilizando el algoritmo de metrópolis y el Método de Monte Carlo.*

### **1. Introducción**

*En **primer lugar**, se pedía estimar la posición del centro de masa de un sólido (y su error asociado) descrito por la intersección de un toro y un cilindro dados por las siguientes ecuaciones:*

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

*Con la densidad del sólido dada por la siguiente fórmula:*

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

*Para resolver el problema usando alguno de los métodos de integración de Monte Carlo (el cual era el objetivo de este problema) se tenía que definir un volumen lo más pequeño posible que contuviese al sólido entero sobre el cual se quería integrar.*

*En **segundo lugar**, se pedía obtener una muestra aleatoria de números con la distribución (no normalizada) dada por la siguiente ecuación:*

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

*Para obtener la muestra aleatoria había que utilizar el algoritmo de Metrópolis con una distribución de proposición  $x_p = x_n + \delta * r$ , donde  $r$  era una variable aleatoria de la distribución uniforme  $U(-1, 1)$ . La variable  $\delta$  tendría un valor fijo que se debería determinar. A modo de consejo, se afirmaba además, que un buen valor era aquel para el cual se aceptaban aproximadamente 50% de las proposiciones.*

*Junto con esto, se añadía el tener que generar una muestra de unos 10 millones de puntos. Y para comprobar que el resultado fuese adecuado se graficaba  $W(x)$  y un histograma de las variables aleatorias, ambos apropiadamente normalizadas.*

*Y, para finalizar, se requería determinar la incertidumbre asociada a cada bin del histograma y graficar el histograma con las barras de error asociadas.*

*La estrategia sugerida para hacer esta parte era realizar una simulación estilo Monte Carlo: repetir muchas veces ( $N = 100$  o más) el procedimiento completo del problema, pero cada vez utilizar una semilla distinta, o un punto de partida distintos o ambos. El resultado sería que se obtendrían  $N$  histogramas que deberían ser distintos debido a la naturaleza aleatoria del procedimiento. Para un bin dado, con  $N$  valores distintos, el tamaño de la barra de error se igualó a la desviación estándar de los  $N$  valores para ese bin.*

## **2.Procedimiento**

### **Parte 1**

#### **“ Calculando el centro de masa de la intersección del toro y cilindro”**

*Se definieron en primer lugar, las funciones asociadas a la densidad en cualquier punto del espacio junto a las ecuaciones del toro y del cilindro.*

*La utilidad de las ecuaciones del toro y del cilindro, era que se utilizarían para señalar qué puntos se encontraban dentro o fuera de la intersección. De esta forma, el paquete de puntos  $(x,y,z)$  que no lo cumpliera no entrarían en la*

*iteración impuesta para aplicar el Método de Monte Carlo.*

*La finalidad de aplicar el Método de Monte Carlo radicaba en que dados los puntos (x,y,z) aleatorios formados y que entrasen en la iteración, posteriormente, se aplicaría la definición de centro de masa para calcular cada componente (X<sub>cm</sub>,Y<sub>cm</sub>,Z<sub>cm</sub>) por separado.*

*Dicho esto, la ecuación ocupada para el cálculo del centro de masa fue:*

$$X_{cm} = \frac{\sum_{i=1}^N (\rho(x, y, z) * x)}{\sum_{i=1}^N (\rho(x, y, z))}$$

$$Y_{cm} = \frac{\sum_{i=1}^N (\rho(x, y, z) * y)}{\sum_{i=1}^N (\rho(x, y, z))}$$

$$Z_{cm} = \frac{\sum_{i=1}^N (\rho(x, y, z) * z)}{\sum_{i=1}^N (\rho(x, y, z))}$$

*siendo*  $\vec{R}_{cm} = (X_{cm}, Y_{cm}, Z_{cm})$

*A su vez, para realizar este proceso, se eligió un volumen con los siguientes límites:*

- X entre 1 y 4*
- Y entre -4 y 4*

- Z entre -1 y 1

Los motivos por los que se definieron esos límites fueron porque de la ecuación toroidal y del cilindro se podía obtener fácilmente que  $|z| < 1$  ( el radio del cilindro era 1 de hecho y el toro tenía el mismo intervalo para z) . Por otro lado, para el caso de x, al tener un cilindro de centro  $c(x,z) = (2,0)$  con radio=1, se podía apreciar que existía un “desplazamiento” de x, y además, como se cumplía que  $x \geq 1$  en la ec. del cilindro y, de la ec. del toro que  $R=3$  y  $r=1$ , se extendió el valor de x desde 1 hasta 4. Ahora, para el caso de y, reemplazando estas condiciones en la ec. Toroidal, se obtenía que y debía pertenecer al intervalo  $[-4,4]$ .

De esta forma, luego, se decidió por usar  $N=100000$  puntos  $(x,y,z)$  para calcular el centro de masa asociado a una semilla en particular. Así, con tal de definir estadísticamente un valor para la posición del centro de masa, es que se calcularon 120 centros y, simplemente, con estadística se lograron obtener los errores de cada medida junto a un valor aproximado del centro de masa.

## **Parte 2**

### **“ Distribución $W(x)$ y su desarrollo con el Algoritmo de Metrópolis”**

Para esta parte se ocupó el Algoritmo de Metrópolis, que básicamente señala que dada una distribución de probabilidad  $W(x)$  conocida ( no necesariamente normalizada) es posible generar una secuencia  $\{x_i\}$  de valores aleatorios que se ajusten a esta distribución mencionada.

Los pasos a realizar para el algoritmo eran los siguientes:

1) Asumir un  $x_n$  ( se asumió  $x_n=0$  para este problema, ya que se concentra cierta parte de la distribución de probabilidad en este punto)

2) Generar un conjunto de números aleatorios nuevos  $x_p$  ( de prueba)

3) Probar estos numeros con el criterio de Metrópolis y decidir si se aceptan o no :

si  $x_p$  se aceptaba ---->  $x_{n+1}=x_p$

si  $x_p$  se rechazaba ---->  $x_{n+1}=x_n$

donde el criterio de Metrópolis señalaba que :

si  $w(x_p)/w(x_n) > r$  (  $r$  distribuye uniformemente entre 0 y 1) ---->  $x_p$  se acepta, caso contrario no.

Para realizar esto, se realizó el cálculo numérico con tal de que la probabilidad de pasar de  $x_n$  a  $x_p$ , o de  $x_p$  a  $x_n$  sea equiprobable, es decir, se consideró aceptar un 50% de los valores propuestos para fuesen apropiadas las mediciones.

Dicho esto, se calculó inicialmente la integral de la distribución  $W(x)$  entregada y se procedió a normalizarla ( para no tener inconvenientes con el gráfico de la función dada y el histograma ). Se escribió la función propia del algoritmo de metrópolis y, lo más importante es que para calcular el  $d$  apropiado, con tal de que se aceptarán 50% de los valores de  $x$  propuestos, se crearon 2 funciones: “create” y “calc\_d”, las que permitieron graficar el porcentaje de valores aceptados v/s el valor de  $d$ . De esta forma se pudo determinar un valor de  $d=3.96$ . ( con una probabilidad de 50.15850158501585% de valores propuestos aceptados)

Con  $d$  calculado y tomando  $x[0]=x_n=0$  y alrededor de 1 millón de puntos ( no se pudo 10 millones por problemas de tiempo de ejecución computacional), ya se tenía lo necesario para graficar el histograma y la distribución de probabilidad dada  $W(x)$  normalizada.

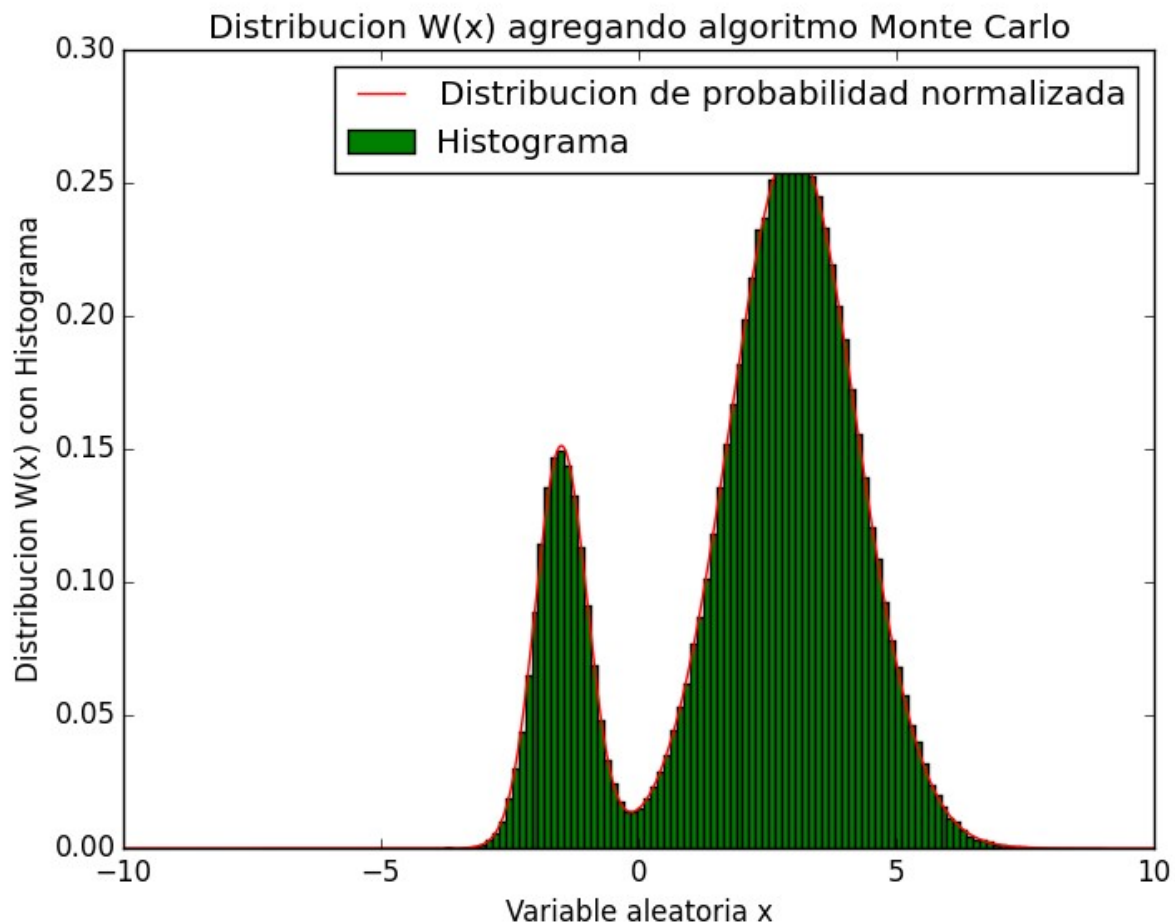
Ahora, para la **parte extra**, se creó una nueva función, parecida a la del algoritmo de metrópolis previamente creada, sin embargo, ésta función retornaría los bins además de los valores(resultados) de la distribución de probabilidad para cada histograma. Se señala esto, ya que se fijaron los valores para los bins(100) entre -5 y 10, mientras que la cantidad de histogramas para iterar fue de 100 también, y la cantidad de puntos  $x$  a trabajar fue de 10000 y no más ,debido a demoras por el tiempo de ejecución del programa para numeros mayores ( optimización ).De esta forma, es que para cada bin (de cada histograma) habría un promedio y desviación estandar asociado, con el que aplicado a los 100 histogramas, se logró obtener una buena aproximación (estadísticamente hablando) de los errores para cada bin, los que en este caso, se consideraron como la desviación estándar.

### **3. Resultados**

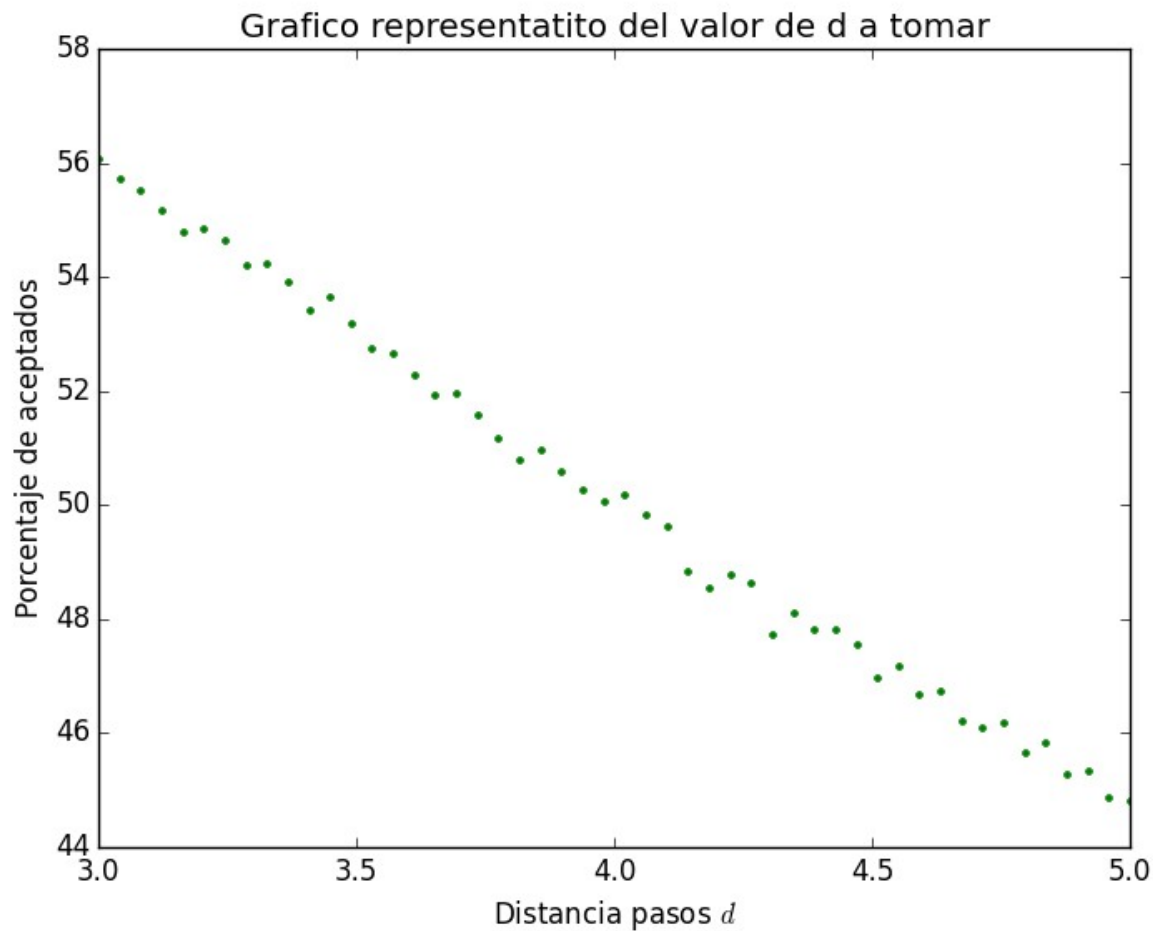
A continuación se anexan los resultados obtenidos para la parte 2 .Además se añaden los valores del centro de masa obtenidos mediante estadística de la parte 1.

Centro de masa	Xcm	Ycm	Zcm
Posición	2.080	0.001	0.000
Error	0.003	0.015	0.003

**Tabla1.** Valores de la posición del centro de masa en cada coordenada (unidades arbitrarias) para la parte 1, anexando los errores respectivos de cada componente.

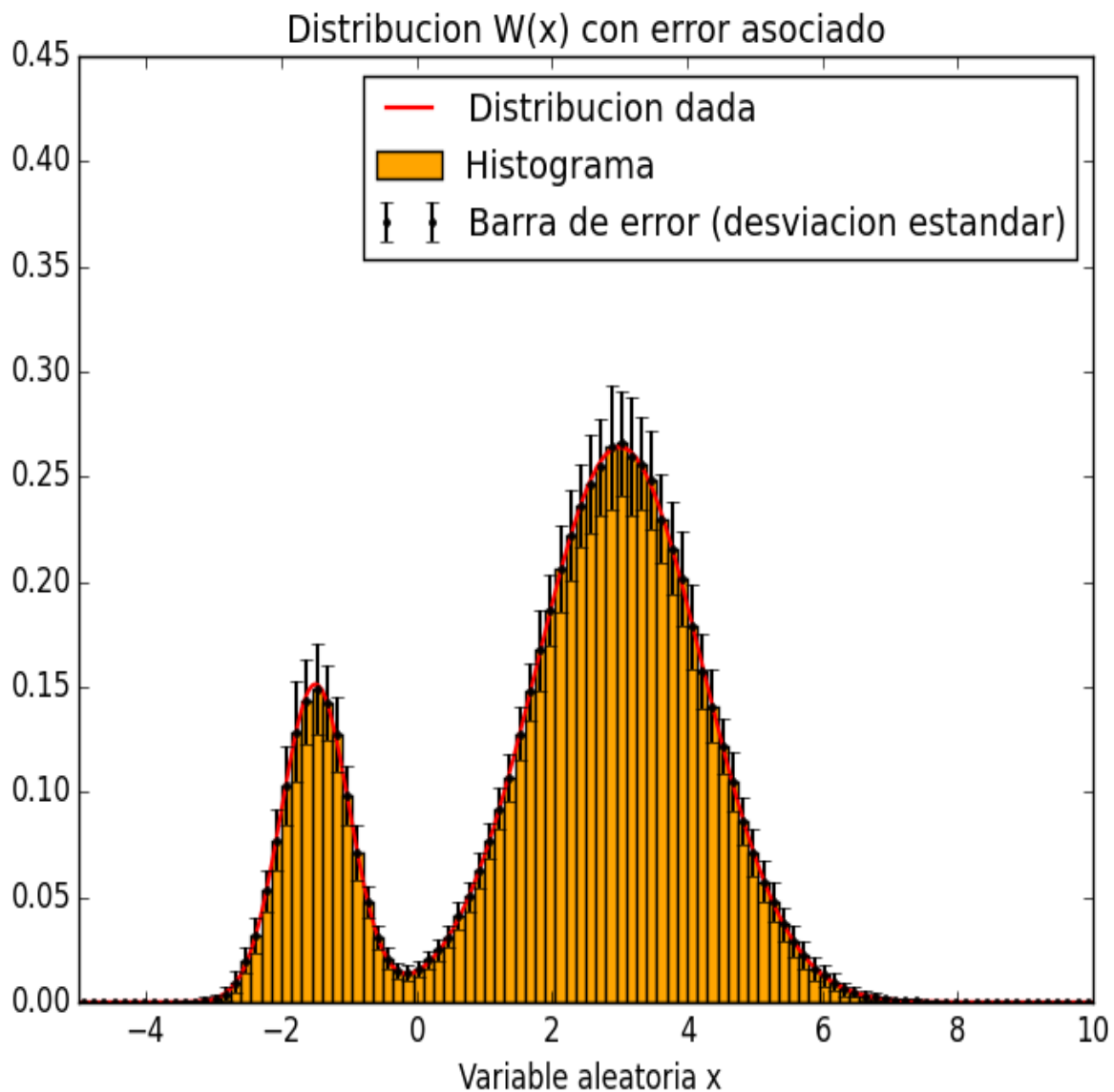


**Figura1.** Gráfico de la Distribución de probabilidad  $W(x)$  dada normalizada en color verde junto al histograma creado mediante el algoritmo de metrópolis v/s la variable aleatoria  $x$ . Corresponde a la parte 2 con  $d=3.96$  y  $x_n=0$ .

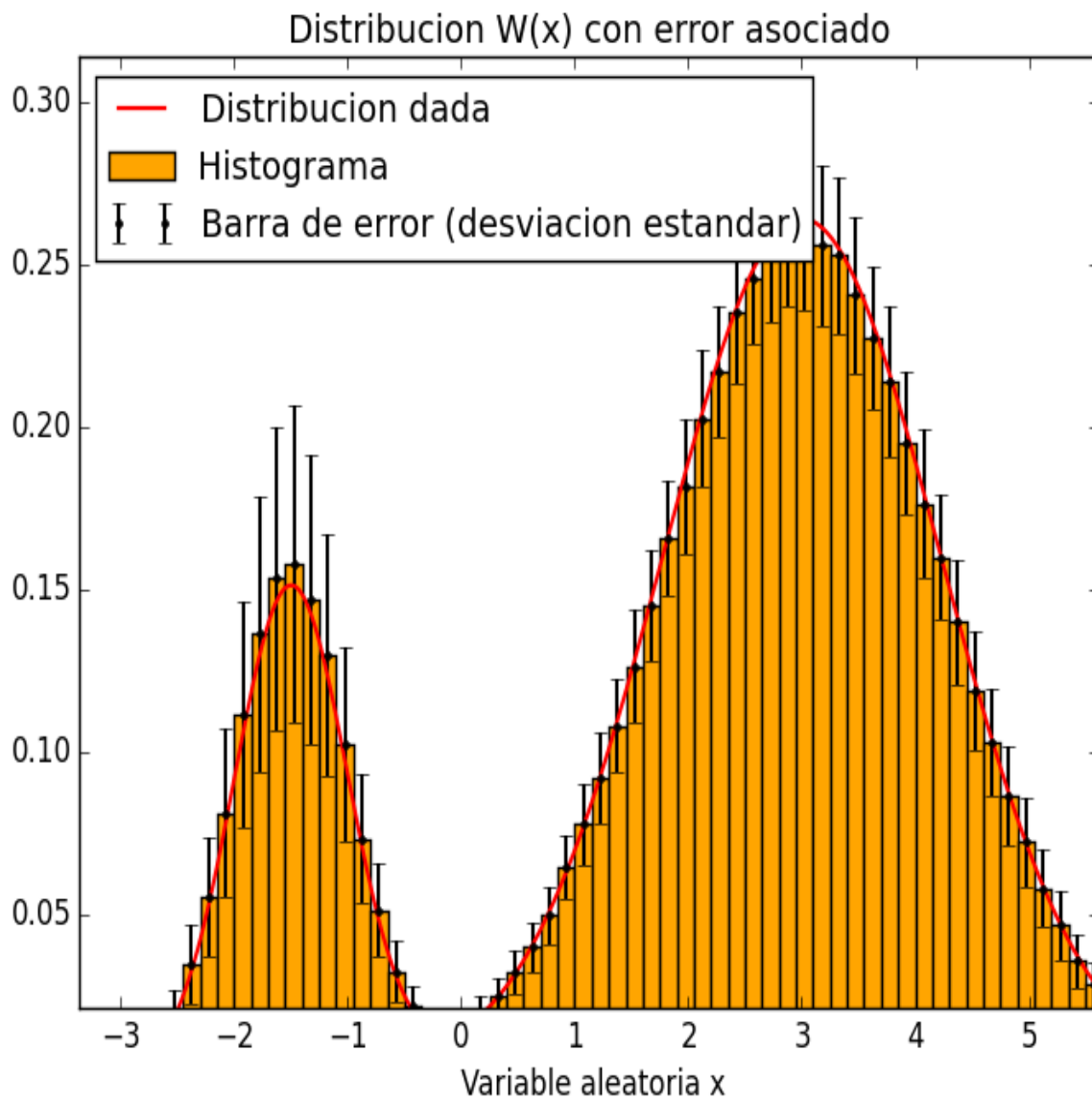


**Figura2.** Gráfico de  $d$  ( distancia a tomar para el algoritmo de metrópolis ) v/s el porcentaje de valores propuestos aceptados. Se encontró un valor aproximado de  $d=3.96$  que se consideró apropiado para el problema, puesto que la probabilidad para valores propuestos aceptados alcanzó un 50.158 %. ( de un total de 100000 puntos)

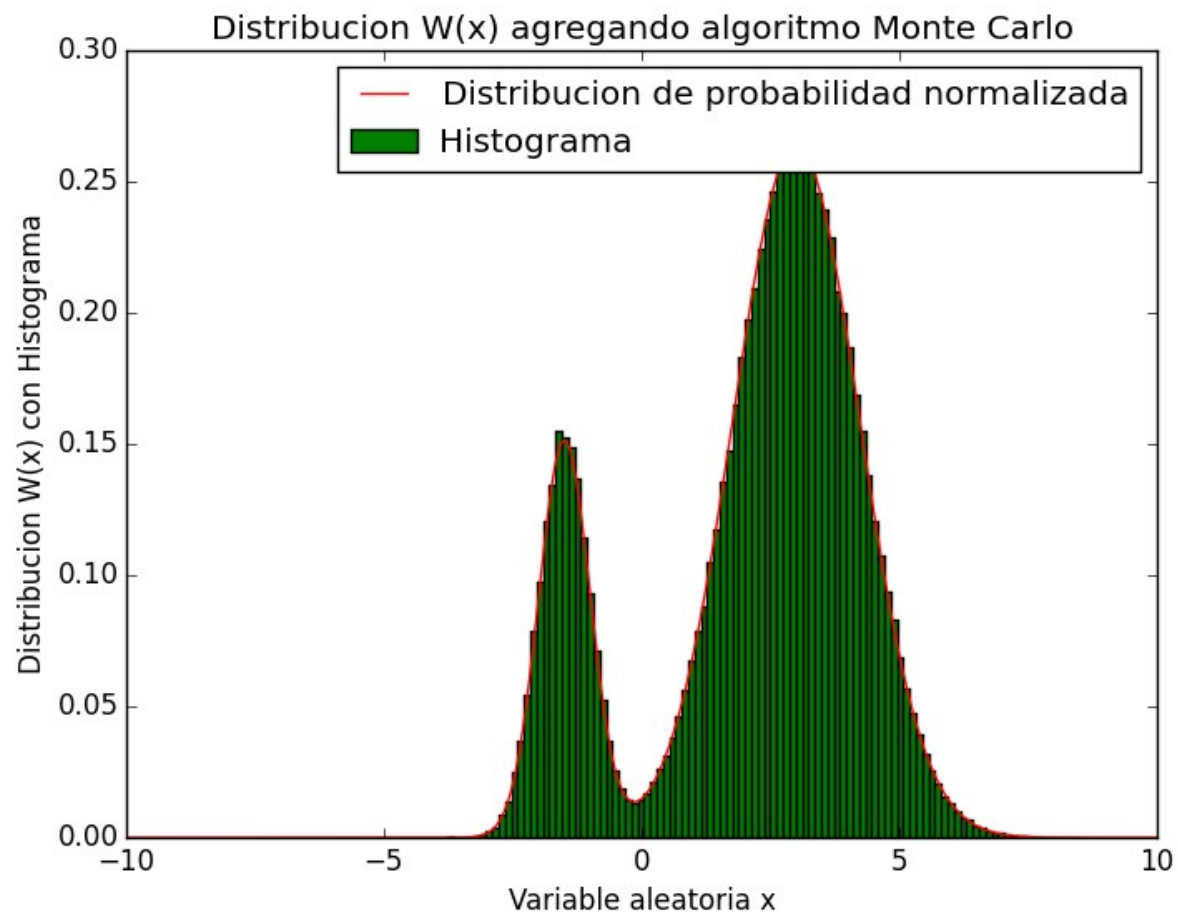




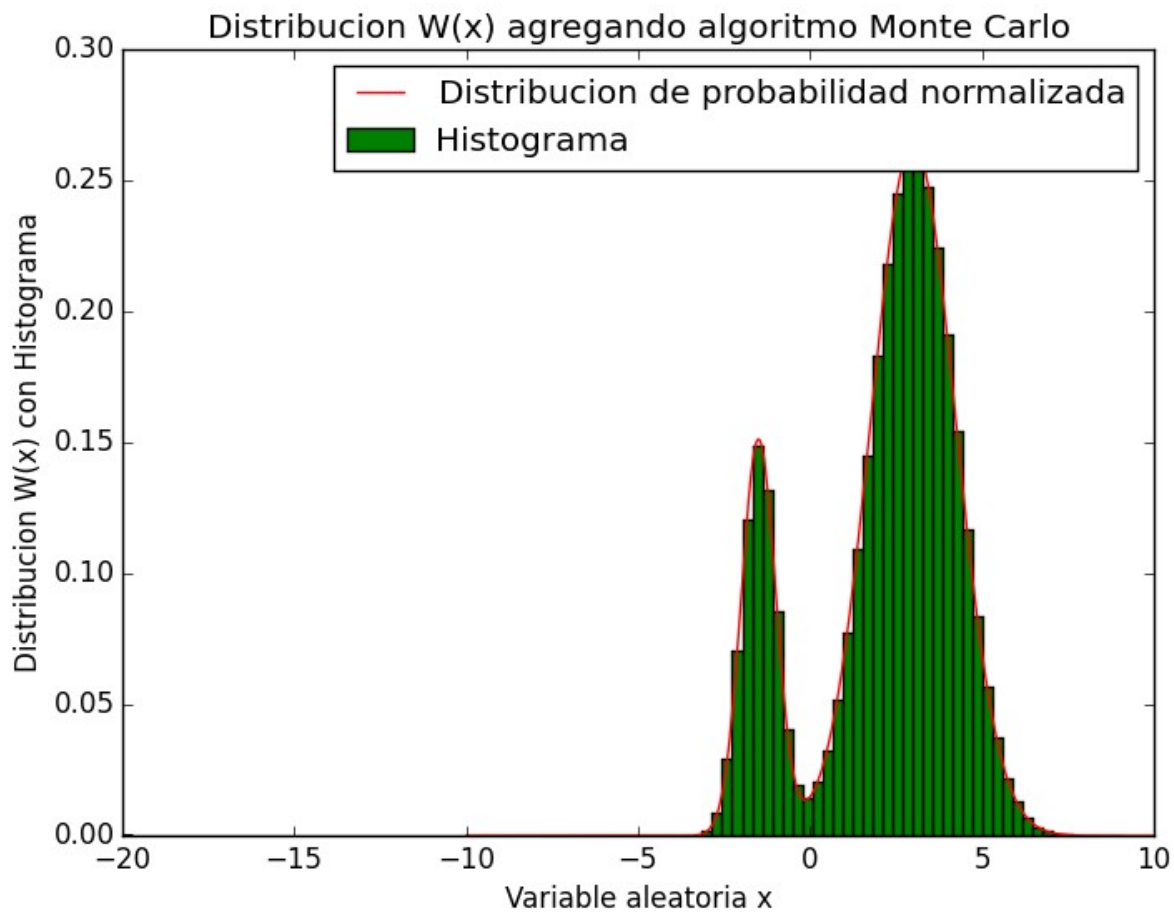
**Figura3.** Gráfico de la distribución  $W(x)$  dada normalizada junto al histograma promedio de los 100 tomados v/s la variable aleatoria  $x$  ( $d=3.96$  y  $x_n=0$ ) para la parte 2 extra .Se puede notar que cada bin presenta un error que viene dado por la desviación estándar.



**Figura4.** Gráfico de la distribución  $W(x)$  dada normalizada junto al histograma promedio de los 100 tomados v/s la variable aleatoria  $x$  ( $d=1$  y  $x_n=0$ ) para la parte 2 extra. Se puede notar que cada bin presenta un error que viene dado por la desviación estándar.



**Figura5.** Gráfico de la Distribución de probabilidad  $W(x)$  dada normalizada en color verde junto al histograma creado mediante el algoritmo de metrópolis v/s la variable aleatoria  $x$ . Corresponde a la parte 2 con  $d=1$  y  $x_n=0$ .



**Figura6.** Gráfico de la Distribución de probabilidad  $W(x)$  dada normalizada en color verde junto al histograma creado mediante el algoritmo de metrópolis v/s la variable aleatoria  $x$ . Corresponde a la parte 2 con  $d=3.96$  y  $x_n=-20$ . Se desplaza hacia la derecha ya que se parte desde el punto  $x=-20$ .

## **4. Conclusión**

Se pudo concluir que para la **primera parte**, al resolver el problema del sólido que se formaba por la intersección de la ecuación del toro y del cilindro, las coordenadas  $(X_{cm}, Y_{cm}, Z_{cm}) = (2.080, 0.001, 0.000)$  obtenidas se podrían haber intuido previamente. Analizando, el cuerpo sólido, por simetría se podía encontrar que tanto  $Y_{cm}$  como  $Z_{cm}$  eran muy cercanos a cero. Sin embargo, a la hora de haber tratado de obtener una aproximación para  $X_{cm}$ , no se lograba con facilidad. Esto pudo ser debido a que la densidad entraba más en consideración en esta componente que en las 2 restantes. Y además, dado que no existía una simetría explícita, también hacía más complicado el intuir el punto.

Por otro lado, cuando se vieron los errores que arrojaban los puntos generados, se podía ver que efectivamente para  $Y_{cm}$  el error era más grande (0.015), lo cual se relacionaba directamente con que esta coordenada consideraba mayor rango de valores  $(-4, 4)$  que las demás coordenadas, lo que originaba que a la larga, existiesen más posibilidades de regiones en el espacio 3D donde se podría haber encontrado el centro de masa asociado a  $Y$ . Para el caso de  $X_{cm}$  y de  $Z_{cm}$ , el error era bastante más pequeño (0.003 para ambos), de hecho, terminó siendo  $1/3$  del error asociado a la componente  $Y$ , lo cual se explica porque el intervalo de ambos ejes se encontraba más acotado que el de  $Y$ . Sin embargo, de todas formas aunque haya sido casi despreciable, igual hubo un error mínimo para estos ejes. ( ver tabla 1 anexada)

Así, si analizando el Método de Monte Carlo para la primera parte, pudimos deducir que efectivamente fue un buen algoritmo numérico, que a primera instancia permitió calcular valores con alta precisión, siempre y cuando, se trabajaran con muchos puntos ( en este caso unos 100000 puntos para calcular cada vector posición centro de masa). De todas formas, esto no siempre es lo más idóneo, debido a que el tiempo de ejecución termina siendo muchas veces una incomodidad para el programador, independiente de que sea preciso el método. Depende de lo que se requiera. Precisión o tiempo.

Para la **segunda parte**, si analizamos la figura 2, nos encontramos con que el valor  $d$  escogido para el Algoritmo de Metrópolis fue una buena medida, ya que sirvió para graficar de manera apropiada los 2 gráficos de las figuras 1 y 3, sin que los histogramas sobresalieran de la función  $W(x)$  normalizada dada por el enunciado. Cabe destacar, que el hecho de haber escogido un valor apropiado era

crucial, puesto que el no tomar un valor apropiado significaba obtener gráficos más imprecisos, tales como la figura 4 y 5, donde al elegir  $d=1$  con  $x_n=0$ , las barras de los histogramas empezaban a sobresalir de la función  $W(x)$  normalizada. Esto sucedió, de manera análoga al haber escogido un valor de  $x_n$  muy lejano en cuanto a la concentración de la función distribución de probabilidad ( donde tiende a cero). En la figura 6 se puede apreciar este fenómeno. Se eligió un valor inicial  $x_n=-20$ , donde la función  $W(x)$  es prácticamente nula, y es por esta razón que dado que  $d$  y  $r$  son valores positivos para el algoritmo, le toma bastante “tiempo” a la variable aleatoria  $x$  llegar a la función ( lo que se traduce en que completa con muchos bins sectores que no nos interesaban a priori). Por este motivo, es que hay menos barras en la parte donde la función no se anula ( la mayoría de ellas, en este caso, de las 100, unas 60, se concentraban en los ceros de ésta ).

Y para concluir, en de la parte de 2 extra, se pudo inferir que iterando para 100 histogramas, el error asociado a cada bin ( desviación estadar) iba cambiando dependiendo del punto en donde se encontraba la función, y que, era mayor a medida que se llegaba a máximos locales y a puntos extremos de la función, lo que se podría interpretar como que el grado de incertidumbre era mayor. Sin embargo, a grandes escalas, el error con respecto los histogramas no era considerable. Y el Método de Monte Carlo ( repetido 100 veces) funcionó de manera adecuada.