Cuarto informe de Física Computacional: Números aleatorios y aplicaciones

Francisco Fernández

Mayo del 2020

Resumen

En este laboratorio se analizaron los distintos generadores de números aleatorios en una aplicación de interés física como la caminata al azar y en integrales de Monte Carlo. En el primer caso se estudió el desplazamiento cuadrático medio en función del número de pasos y la cantidad de experimentos. En el segundo el error absoluto de la integral de una función potencial para distintas cantidades de evaluaciones de la misma y se comparó con el método de *importance sampling*.

1. Introducción teórica

1.1. Números aleatorios

Se dice que un número es aleatorio si la variable u que lo toma no puede predecir que valor tendrá en cada caso en concreto, pero cada uno ellos tiene asociada una probabilidad. Cuando se generan estos números aleatorios con la computadora, en realidad, uno está en presencia de números pseudo–aleatorios, que, a pesar de ello, pueden ser utilizados para simular problemas físicos de naturaleza estocástica.

Podemos definir una secuencia de números aleatorios cuando no hay correlación entre ellos y, además, si todos los valores posibles tienen la misma probabilidad se dice que la secuencia es *uniforme*. Los generadores de los cuales se hablará cumplen con esta propiedad y están distribuidos entre 0 y 1, ya que a partir de esta se pueden obtener distintas distribuciones.

1.1.1. Generadores

Existen distintos tipos de generadores de números aleatorios. El más común de ellos es el *generador congruencial lineal*, en el que se multiplica un número aleatorio previo por una constante, se le suma otra distinta, se toma el módulo por M y se queda con la parte remanente como el siguiente número aleatorio. Al primer valor de una secuencia se lo conoce como *semilla* y suele ser dada por el usuario. Este método genera valores en un rango que va desde 0 hasta $M-1$, que se dividen por M para llevarlos al intervalo $[0,1)$, y tienen un periodo P asociado al partir del cual se repite la misma secuencia.

Para el caso de los problemas que se analizarán en este informe se utilizaron, principalmente, dos generadores de números aleatorios. El primero de ellos de Marsaglia ($mzran$), que combina

dos generadores distintos obteniendo un período del orden de 2^{94}. El segundo de ellos es el Mersenne-twister, que tiene un período de 2^{19937}.

1.1.2. Distribuciones no uniformes

Método de la función inversa

Este método para obtener distribuciones de números aleatorios no uniformes consiste en encontrar la inversa de la función de probabilidad acumulada de la densidad de probabilidad en cuestión. Entonces, si se tiene que $F(x)$ es una distribución de probabilidad acumulada, y $F^{-1}(y)$ con $y \in [0, 1)$, su inversa. Si se define $X = F^{-1}(U)$, con U una variable aleatoria con distribución uniforme entre $[0, 1)$. Luego, X está distribuida como F, es decir, $\mathcal{P}(X \leq x) = F(x)$, $x \in \mathcal{R}$. La principal desventaja que tiene es que sirve solo para aquella funciones a las cuales se les puede obtener la inversa de forma analítica.

Método de Box–Müller

La distribución Gaussiana es un ejemplo de una función que su integral no es simple de escribir de forma analítica e invertir, por eso este método propone llevar al problema a 2d, generar un radio y un ángulo aleatorio a partir de una distribución uniforme,

$$x_1 = \sqrt{-2 \ln u_1} \cos 2\pi u_2, \qquad x_1 = \sqrt{-2 \ln u_1} \operatorname{sen} 2\pi u_2,$$

de esta manera se obtienen dos números aleatorios distribuidos de forma Gaussiana.

1.2. Caminata aleatoria

Existen muchas formas de simular caminatas aleatorias y se pueden llegar a distintas descripciones físicas asumiendo distintas suposiciones. Acá se opta por describir, brevemente, un caminante en 2d con pasos discretos, que es lo que luego de analizará en la sección (§2). Se asume que cada cierto tiempo característico τ el caminante da un paso y puede ir hacia arriba, hacia abajo o hacia los costados con igual probabilidad, sin depender del paso anterior.

La distancia radial R desde el punto inicial, acá el origen, después de N pasos se puede obtener de la siguiente relación

$$R^2 = \left(\sum_i^N \Delta x_i \right)^2 + \left(\sum_i^N \Delta y_i \right)^2 ,$$

a tiempos largos los términos cruzados se anulan, porque la probabilidad es la misma para todas las direcciones, entonces

$$R^2 = \sum_i^N \left[(\Delta x_i)^2 + (\Delta y_i)^2 \right] ,$$

que es el *desplazamiento cuadrático medio*. Y en nuestro caso, como los pasos son de longitud fija,

$$R^2 \propto N. \tag{1}$$

Si se generaliza esto para d dimensiones y permitimos que los pasos no sean todos iguales, pero su distribución sea tal que su media sea nula, entonces se obtiene la ecuación de difusión de Einstein

$$D = \frac{\langle r^2 \rangle}{2d\tau},$$

donde $\langle r^2 \rangle$ es el segundo momento de los pasos.

1.3. Integración de Monte Carlo

La técnica de Monte Carlo para calcular integrales se basa en el *teorema del valor medio*

$$I = \int_a^b dx f(x) = (b-a)\langle f \rangle ,$$

donde $\langle f \rangle$ es el valor medio de $f(x)$ el el intervalo de integración. El algoritmo de integración consiste en evaluar $\langle f \rangle$ con una secuencia de números aleatorios x_i distribuidos uniformemente en $[a, b)$, entonces se puede estimar

$$f(x) \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i),$$

de donde resulta la regla de integración

$$I \equiv \int_a^b dx f(x) = \frac{b-a}{N} \sum_{i=1}^{N} f(x_i).$$

Generalizando a más dimensiones resulta, de una manera directa,

$$I_N \equiv \int_V d\mathbf{x} f(\mathbf{x}) \approx \frac{V}{N} \sum_{i=1}^{N} f(\mathbf{x}_i), \tag{2}$$

donde V es el volumen de integración. I_N es una suma de variables aleatorias entonces, cuando N es grande la distribución tiende a una Gaussiana (*teorema central del límite*), cuya varianza viene dada por

$$\sigma^2_{I_N} = \frac{V^2}{N} \sigma^2_f,$$

por lo tanto,

$$I = I_N \pm \frac{V}{\sqrt{N}} \sigma_f.$$

De esta última expresión se puede ver que el error del método va como $\sim \frac{1}{\sqrt{N}}$, y que él mismo no depende de la dimensión de la integral.

1.3.1. Muestreo de importancia (*importance sampling*)

Este algoritmo sirve para mejorar la convergencia ya que permite, a través de una función de peso, evaluar el integrando en las regiones más importantes. Esto se deriva de expresar la integral de la siguiente forma

$$I = \int_a^b dx \frac{f(x)}{w(x)} w(x),$$

donde $w(x)$ es la función de peso o la probabilidad de distribución de nuestros números aleatorios. La aproximación de MC con muestreo de importancia resulta

$$I \approx I_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{w(x_i)}.$$

Si la integral de $w(x)$ tiene inversa analítica, W^{-1}, se puede usar el método de inversión y obtener

$$I_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(W^{-1}(u_i))}{w(W^{-1}(u_i))},$$

con u_i distribuidos entre $[0, 1)$.

1.3.2. Método de rechazo

Este algoritmo consiste en encontrar una función $f(x)$ tal que tenga área finita A y sea mayor o igual a la distribución de probabilidad $p(x)$ para todo x. La integral de $f(x)$, F, debe ser invertible. Si se genera un número aleatorio u en $[0, A)$, obtener $x = F^{-1}(u)$, generar un número aleatorio y uniforme en $[0, f(x))$ y x se acepta si $y < p(x)$, si no se rechaza.

2. Resultados y discusiones

2.1. Caminata al azar

En el siguiente problema se analiza una caminata al azar en una red cuadrada bidimensional. En la figura (1) se observan diez trayectorias distintas, todas comenzando en el origen y con 1000 pasos. Vemos como la generación de distintos números aleatorios hace que el caminante tome direcciones distintas, a veces quedándose cerca del origen, al rededor de 20 longitudes de paso y otras yéndose a algún cuadrante en particular, estas distintas opciones se analizarán más adelante.

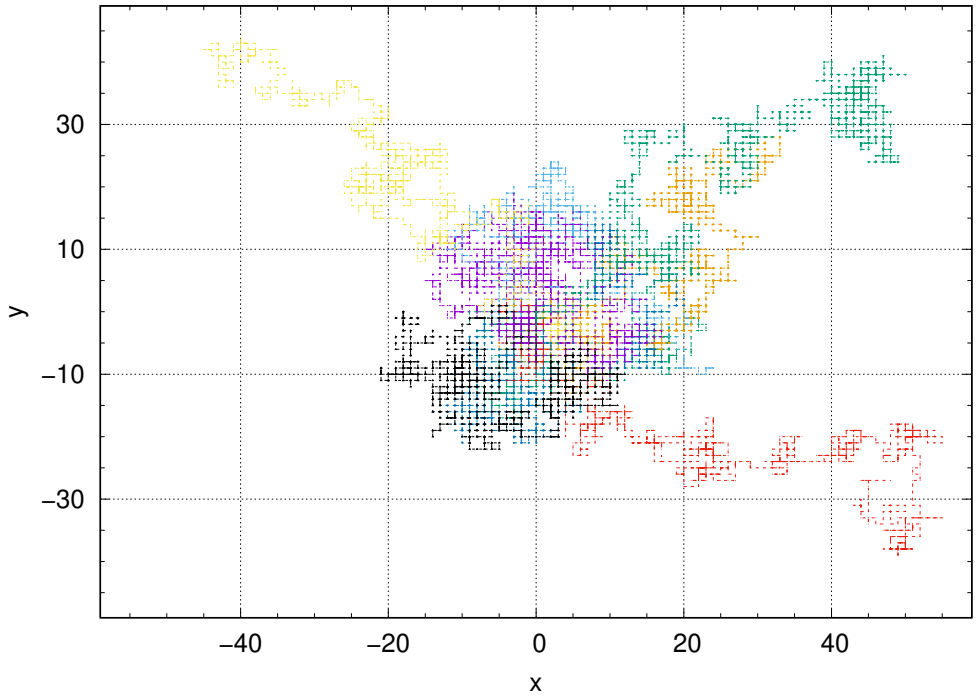


Figura 1: 10 caminatas aleatorias en el plano xy comenzando en el origen.

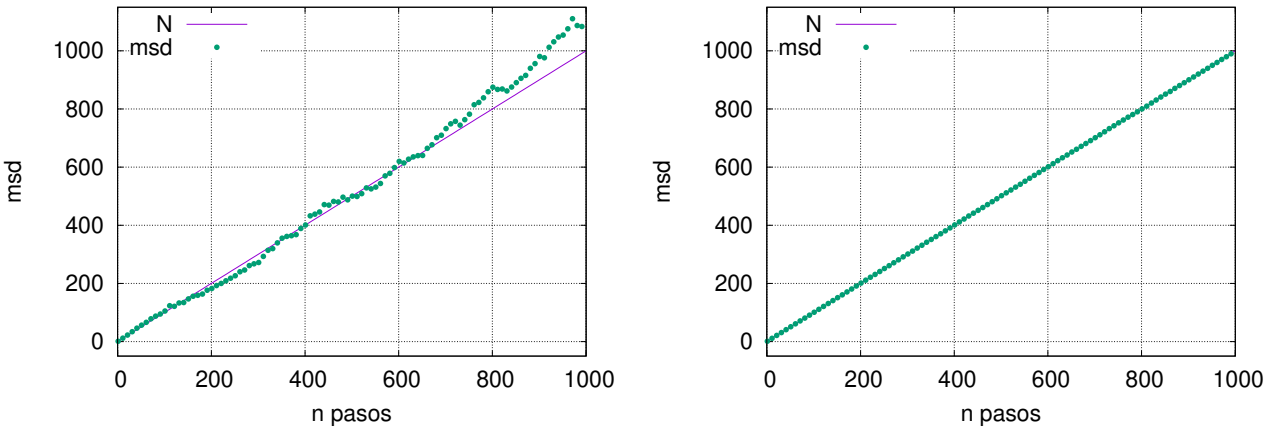
Para este problema se juntaron los generadores *ran*0, escrito con ayuda del teórico, *ran*2 de *Numerical Recipies*, *mzran* de Marsaglia y el *Mersenne–Twister* en un módulo `randomnum.f90` para poder acceder a todos ellos e ir cambiando rápidamente con solo remplazar el puntero que se encuentra al principio de los distintos *mains*.

2.1.1. Inciso a: desplazamiento cuadrático medio

Para este inciso se escribió la caminata con un if con cuatro condiciones distintas para el número aleatorio y se almacenaron los distintos valores del desplazamiento cuadrático medio (*msd* de sus siglas en inglés, *mean square displacement*) en distintas entradas de un vector según el paso, para todos los experimentos realizados. Luego se realizó el promedio sobre la

cantidad de experimentos y se escribió en un archivo de salida. Lo descripto se encuentra en `2d-msd.f90`.

En la figura (2) observamos dos casos distintos del desplazamiento cuadrático medio en función de la cantidad de pasos. En el primero de ellos (figura 2a) vemos como, para 100 experimentos de 1000 pasos cada uno de ellos, el desplazamiento cuadrático medio sigue ligeramente la relación descripta por la ecuación (1), para pasos largos se ve como empieza a estar por encima de esta predicción. En el caso en que se realizaron 10^5 experimentos (figura 2b) se ve como el msd es directamente proporcional a la cantidad de pasos n. Cabe destacar que este comportamiento lo esperaba para tiempos largos, ya que ahí realmente se cancelan los términos intercalados del msd, por lo cual se puede concluir que realizar un solo experimento infinito es equivalente a realizar una gran cantidad de experimentos (10^5) cortos (10^3 pasos) y promediar sobre ellos.



(a) 100 experimentos.

(b) 100000 experimentos.

Figura 2: Desplazamiento cuadrático medio en función del número de pasos n.

2.1.2. Inciso b: cuadrantes

En este inciso se utilizó la misma idea que en el anterior sólo que en vez de obtenerse el msd se analizó la cantidad de veces que el caminante se encontraba en algún cuadrante, asignando un semieje a cada uno de ellos, dejando solo fuera el origen. Este programa se encuentra en `2d-cuadrante.f90`.

En la figura (3) se presentan los resultados correspondientes a 10^5 caminatas de 100 pasos. Lo primero que se puede observar es que para valores impares de la caminata, el caminante no puede volver al origen, por lo cual sí o sí se encuentra en alguno de los cuadrantes (probabilidad igual a 0,25); por otro lado, para los pasos pares, como sí puede volver, entonces la probabilidad de que esté en algún cuadrante es menor que $0,25$. Sin embargo, se ve que a medida que la cantidad de pasos aumenta la probabilidad de estar en el origen va tendiendo a cero y se puede observar como para n par los valores de la probabilidad tienden asintóticamente a $0,25$. Por último, puede destacarse que no hay diferencia en los valores obtenidos con los distintos generadores de números aleatorios que se probaron, por lo cual, para este caso, cualquiera de ellos es óptimo, pero si se aumenta el número de experimentos o de pasos hay que tener cuidado con el período de los mismos para que no haya correlaciones.

2.1.3. Inciso c: 3d desplazamiento cuadrático medio

Si se piensa en una partícula real en 3 dimensiones (por ejemplo una molécula aromática que se difunde en una habitación) el modelo propuesto de una grilla no parece ser el más conveniente,

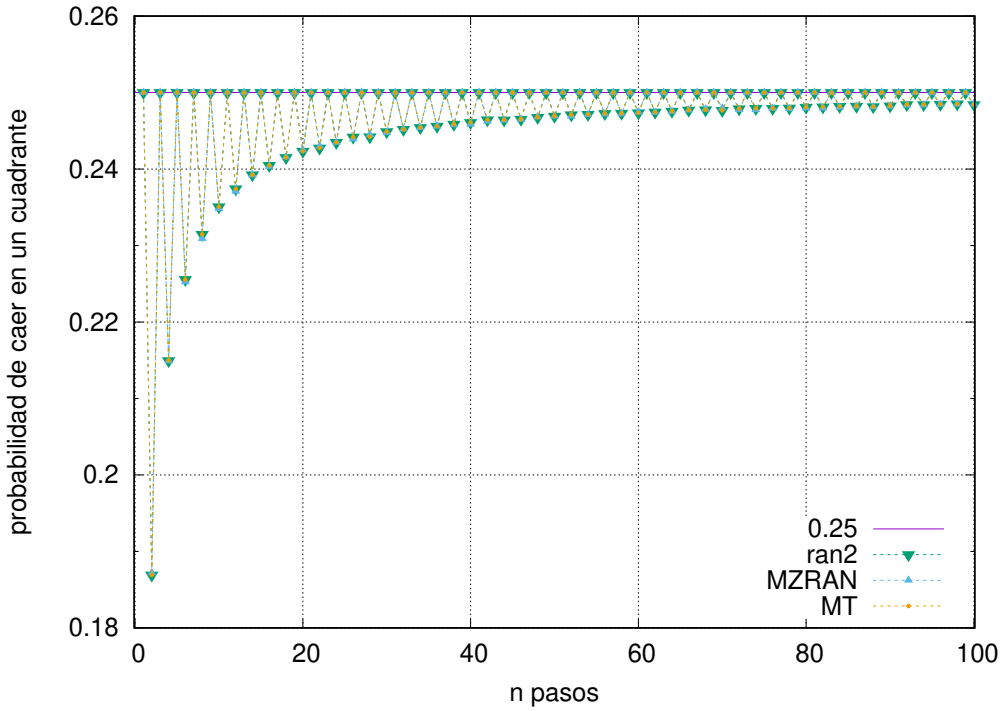
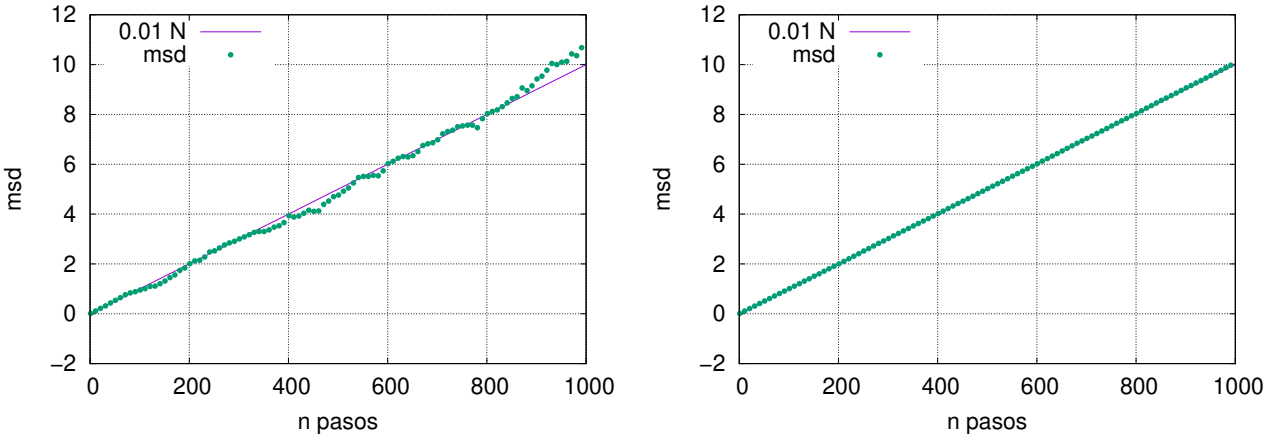


Figura 3: Probabilidad promediada de los cuatro cuadrantes, de que el caminante se encuentre en uno de ellos, en función de n.

por lo que se decidió hacer el paso aleatorio de la caminata en una esfera de radio $0,1$ en cualquier dirección. Para ello se generaron dos números aleatorios y se les realizó la siguiente cuenta,

$$\theta = \arccos(1 - 2\pi u_1), \qquad \phi = 2\pi u_2,$$

de manera que estén distribuidos uniformemente en θ y ϕ. A partir de ellos se encontraron, con coordenadas esféricas, los x, y, z correspondientes. Esto se encuentra en `3d-randomwalk.f90`.



(a) 100 experimentos.

(b) 100000 experimentos.

Figura 4: Desplazamiento cuadrático medio en función del número de pasos n.

Para esta caminata también se muestra, al igual que en el inciso (a), el desplazamiento cuadrático medio en función de los pasos n para la misma cantidad de pasos y experimentos (figura 4). El comportamiento es el mismo que en el caso anterior sólo que la constante, estimada, que acompaña al término lineal en N es dos órdenes de magnitud menor, y esto se debe a la longitud del paso.

2.2. Integración de Monte Carlo

A los módulos utilizados en el problema anterior se sumó el `metodomc.f90`, que tiene dos subrutinas, una del método de Monte Carlo y otra del mismo con *importance sampling*, que toman la función desde el *main* a través de una *interface* en la primera y dos en la segunda, una la función que se quiere integrar y la otra el peso. Para ambos incisos se presentan los resultados obtenidos con el generador de números aleatorios `mzran`.

2.2.1. Inciso a

En este inciso se integró la función $f(x) = x^3$ en el intervalo $[0, 1)$, pero remplazando el valor de la variable *pow* en `a-main.f90` queda generalizado para cualquier potencia mayor que 1. Se realizaron evaluaciones de la $f(x)$ en la expresión (2) hasta $N = 10000$, dando saltos de 10. Los distintos valores obtenidos para la integral fueron comparados con el valor exacto, $I = 1/(n+1)$. En la figura (5) se ve el error absoluto en función de N, en una escala logarítmica. Se graficó la línea $N^{-1/2}$ para observar como la nube de puntos de los errores se dispersa al rededor de la misma variando en distintos ordenes de magnitud según los números aleatorios que hayan sido sorteados y dónde se haya evaluado la función. Se puede decir que el error absoluto para 10000 evaluaciones está al rededor de 10^{-3}.

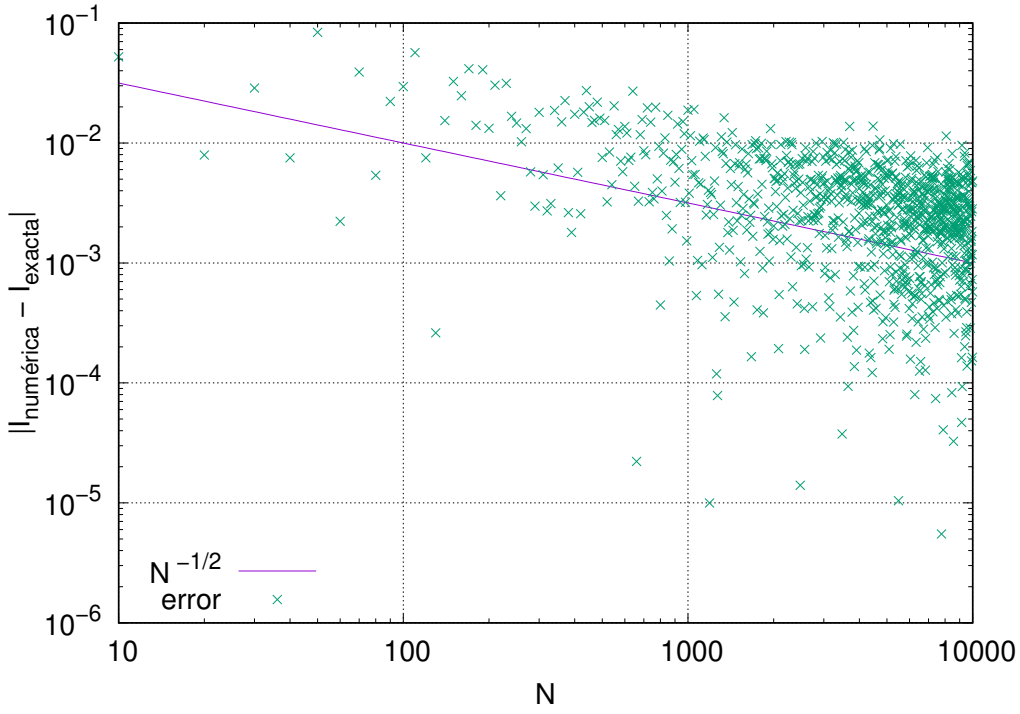


Figura 5: Error absoluto en función del número de evaluaciones de la función.

2.2.2. Inciso b: *importance sampling*

En este caso se obtuvo la integral utilizando *importance sampling*, con una distribución de probabilidad igual a $p(x) = (k+1)x^k$, que se encuentra como segunda función en `b-main.f90`. Para generar números aleatorios distribuidos de esta forma se puede hacer un análisis con el método de la función inversa y llegar a que

$$x = u^{\frac{1}{k+1}}$$

con u distribuidos uniformemente entre $[0, 1)$. Esta distribución está en el módulo `dplpmc.f90`, que es usado por la subrutina de Monte Carlo con importance sampling.

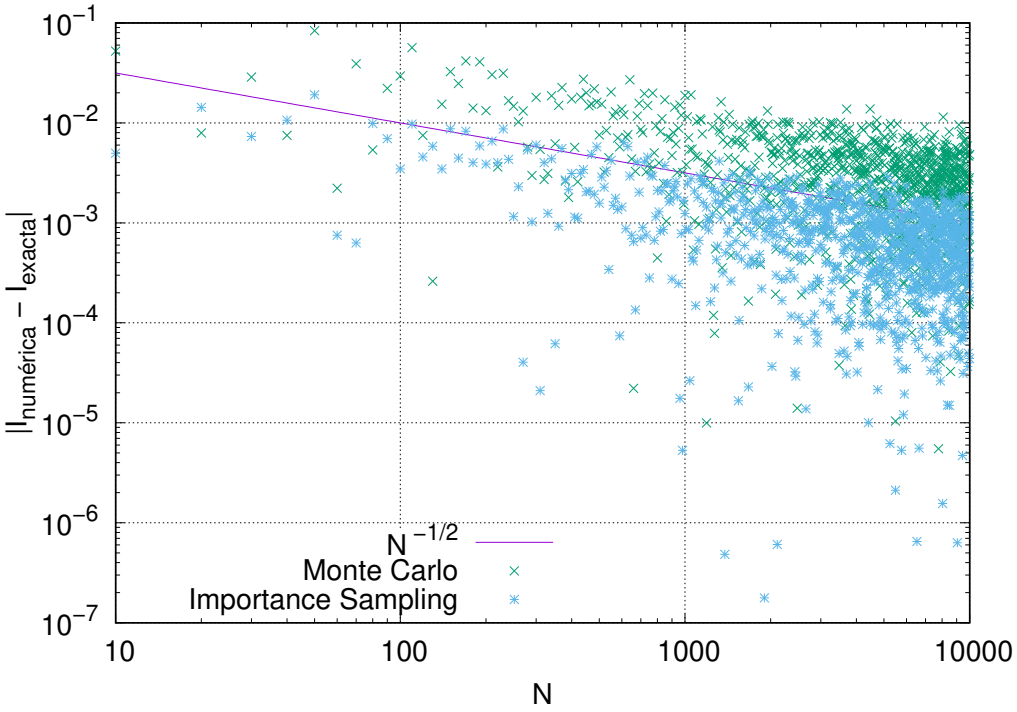


Figura 6: Comparación de los errores absolutos de ambos métodos en función del número de evaluaciones de la función.

Para el caso en el que se eligió $k = 3$, la integral obtenida es exacta con tan solo un paso, pero esto no es de mucha utilidad ya que solo sirve en caso como este en los cuales se conoce la función a integrar. Analizando con más detalle el caso $k = 2$, se observan en la figura (6) los datos obtenidos con este método superpuestos a los del inciso anterior, se ve el mismo comportamiento que antes, la dispersión al rededor de $N^{-1/2}$ pero un orden de magnitud más bajo.