Informe Parte B2: Integración de e^{-x^2} por Monte Carlo

Física Computacional II - Isabel Nieto y Camilo Huertas

10 de julio de 2025

Índice

1. Introducción

El objetivo de esta parte del proyecto es calcular la integral definida de la función $f(x) = e^{-x^2}$ en el intervalo $[0, 1]$ utilizando el método de Monte Carlo por muestreo simple. Se analizará la convergencia del valor estimado de la integral y su error asociado en función del número de muestras N utilizadas en la simulación.

La investigación teórica más amplia sobre el método de Monte Carlo y sus aplicaciones en física estadística se encuentra en el documento principal del proyecto.

2. Fundamento Teórico

2.1. Integración por Monte Carlo (Muestreo Simple)

Dada una integral unidimensional de la forma:

$$I = \int_a^b f(x)dx \tag{1}$$

Podemos reescribirla como el producto del rango de integración y el valor esperado de $f(x)$ si x es una variable aleatoria uniformemente distribuida en $[a, b]$:

$$I = (b-a)\int_a^b f(x)p(x)dx = (b-a)\langle f(X)\rangle \tag{2}$$

donde $p(x) = \frac{1}{b-a}$ para $x \in [a, b]$ y 0 en otro caso.

El método de Monte Carlo estima este valor esperado promediando la función evaluada en N puntos aleatorios x_i, generados uniformemente en $[a, b]$:

$$I_N = (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{3}$$

Por el Teorema del Límite Central, para N grande, I_N se aproxima a I.

2.2. Estimación del Error

El error estadístico (error estándar de la media) de la estimación I_N está dado por:

$$\mathrm{Err}(I_N) = (b-a)\frac{\sigma_f}{\sqrt{N}} \tag{4}$$

donde σ_f^2 es la varianza de la función $f(x)$ en el intervalo $[a, b]$:

$$\sigma_f^2 = \left\langle f^2(X)\right\rangle - (\langle f(X)\rangle)^2 \tag{5}$$

En la práctica, estimamos $\left\langle f^2(X)\right\rangle$ y $\langle f(X)\rangle$ a partir de las muestras:

$$\langle f(X)\rangle \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \overline{f} \tag{6}$$

$$\left\langle f^2(X)\right\rangle \approx \frac{1}{N}\sum_{i=1}^{N} f^2(x_i) = \overline{f^2} \tag{7}$$

Entonces, el error estimado es:

$$\mathrm{Err}(I_N) \approx (b-a)\sqrt{\frac{\overline{f^2} - (\overline{f})^2}{N}} \tag{8}$$

Este error disminuye como $N^{-1/2}$.

2.3. La Integral Específica

La integral a calcular es $I = \displaystyle\int_0^1 e^{-x^2}\, dx$.
Este valor es $\frac{\sqrt{\pi}}{2}\mathrm{erf}(1)$, donde $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$ es la función error.
El valor numérico de referencia es aproximadamente 0,7468241328.

3. Implementación Computacional

3.1. Arquitectura de Software

Se desarrolló una clase `IntegradorMonteCarlo` en C++ que implementa el método de muestreo simple:

- El constructor recibe la función a integrar (como `std::function<double(double)>`), los límites de integración y una semilla para el generador de números aleatorios (`std::mt19937`).
- El método `CalcularIntegralSimple` toma el número de muestras N y devuelve el valor estimado de la integral y el error estándar calculado.

3.2. Programa Principal

El programa principal (`main_montecarlo_integral.cpp`) utiliza esta clase para:

- Calcular la integral de e^{-x^2} para un rango de valores de N (desde 10^1 hasta 10^7)
- Guardar los resultados (N, valor estimado, error estimado, valor teórico, diferencia absoluta) en el archivo `integral_error_Nmax_1e7.dat`
- Facilitar el análisis posterior y la graficación

4. Resultados y Análisis

4.1. Convergencia del Valor de la Integral

La Figura 1 muestra el valor estimado de la integral en función del número de muestras N.
El gráfico demuestra que el valor estimado converge al valor teórico a medida que N aumenta, con fluctuaciones estadísticas que disminuyen progresivamente.

4.2. Convergencia del Error Estimado

La Figura 2 presenta el error estimado de la integral en función de N en escala log-log.
El análisis log-log confirma la dependencia teórica $N^{-1/2}$ del error, característica fundamental del método de Monte Carlo por muestreo simple.

4.3. Diferencia Absoluta con el Valor Teórico

La Figura 3 analiza la diferencia absoluta entre el valor estimado y el valor teórico.
La diferencia absoluta sigue la tendencia general del error estimado, aunque presenta fluctuaciones estadísticas naturales del proceso estocástico.

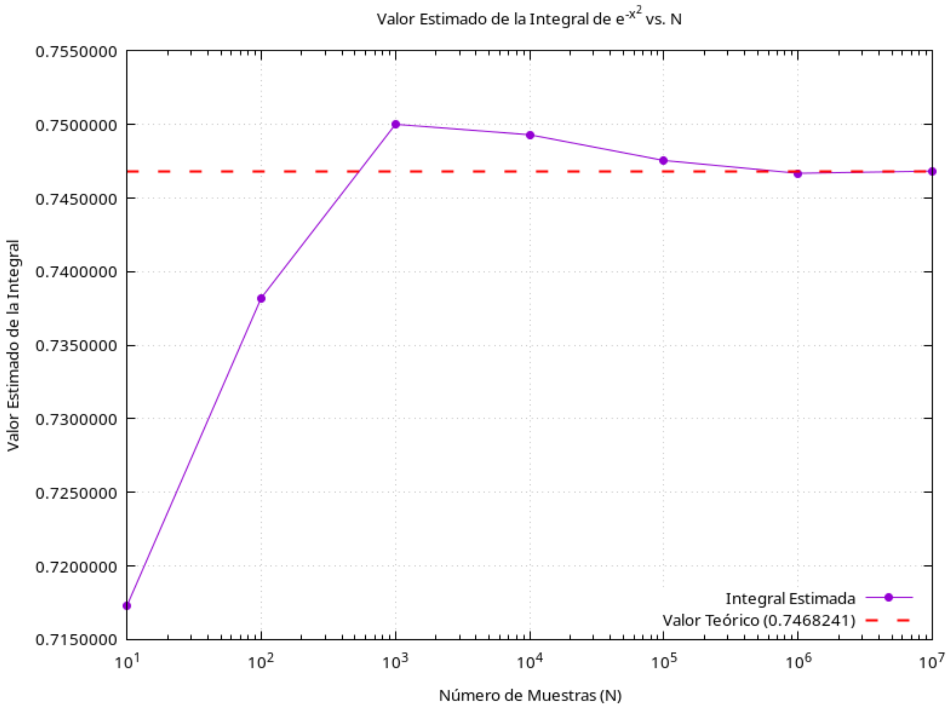

Figura 1: Valor estimado de $\int_0^1 e^{-x^2} dx$ en función del número de muestras N. La línea roja representa el valor teórico ($\approx 0{,}7468$).

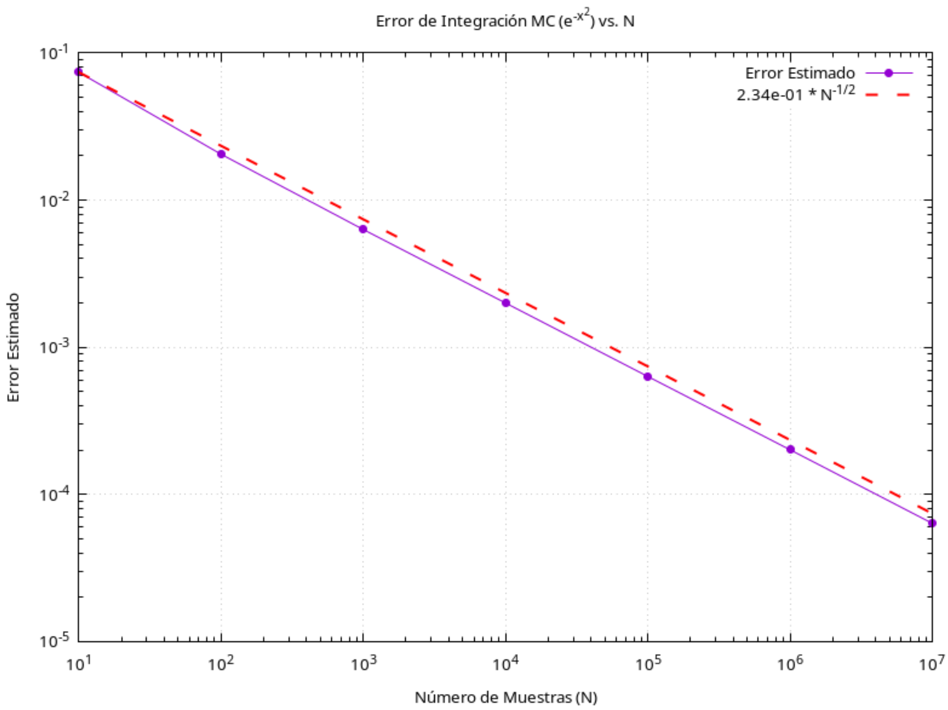

Figura 2: Error estimado de la integral en función del número de muestras N (escala log-log). La línea roja punteada muestra una referencia con pendiente $N^{-1/2}$.

5. Archivos Generados

El programa genera automáticamente:

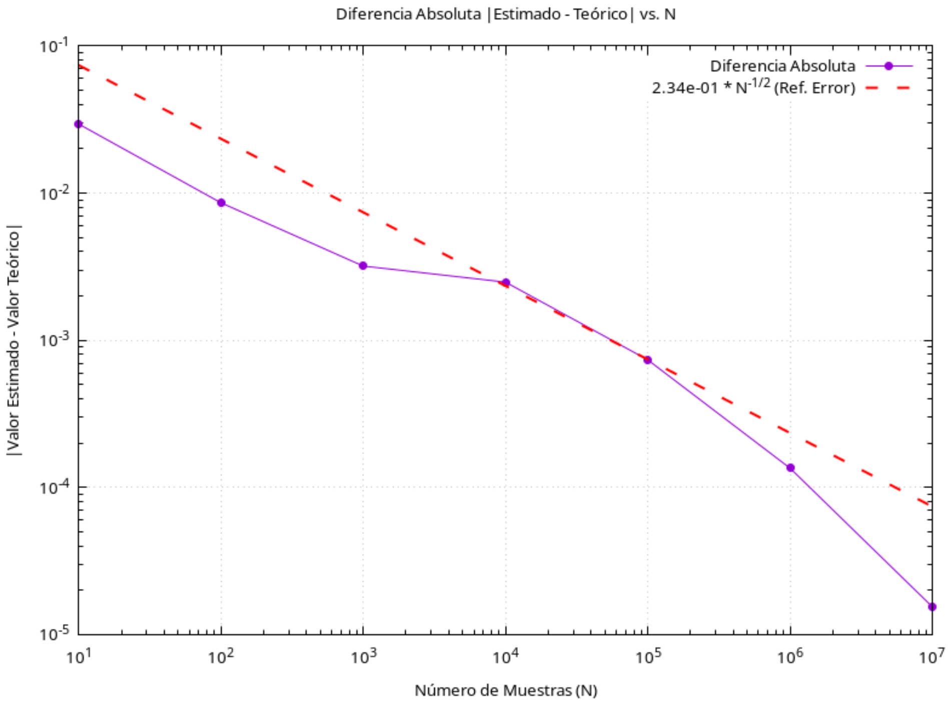

Figura 3: Diferencia absoluta $|I_{estimado} - I_{terico}|$ en función del número de muestras N (escala log-log). La línea roja punteada es la referencia $N^{-1/2}$ del error estimado.

- `integral_error_Nmax_1e7.dat`: Datos de convergencia completos
- Gráficas PNG generadas por el script `plot_integral_error.gp`
- Documentación HTML y LaTeX mediante Doxygen

6. Validación y Verificación

6.1. Precisión Numérica

- Uso de `std::erf` de la biblioteca estándar para el valor de referencia
- Precisión de punto flotante de doble precisión para todos los cálculos
- Verificación de convergencia para $N = 10^7$ muestras

6.2. Reproducibilidad

- Semilla configurable para el generador Mersenne Twister
- Resultados consistentes entre ejecuciones con la misma semilla
- Documentación completa del flujo de trabajo

7. Conclusiones

7.1. Logros Técnicos

1. Se implementó exitosamente el método de Monte Carlo por muestreo simple

2. La clase `IntegradorMonteCarlo` es reutilizable para otras integrales
3. El flujo de trabajo automatizado facilita la reproducibilidad
4. La documentación con Doxygen mejora la mantenibilidad del código

7.2. Resultados Físicos

1. El método converge correctamente al valor teórico de la integral
2. El error estadístico sigue la ley $N^{-1/2}$ característica de Monte Carlo
3. Las fluctuaciones observadas son consistentes con la naturaleza estocástica del método
4. La precisión alcanzada es adecuada para aplicaciones prácticas

7.3. Relevancia Computacional

Este trabajo demuestra:

- La efectividad del método de Monte Carlo para integración numérica
- La importancia del análisis estadístico en métodos estocásticos
- El valor de la programación orientada a objetos para simulaciones científicas
- La utilidad de herramientas automatizadas para análisis de convergencia

7.4. Aplicaciones y Extensiones

El código desarrollado puede extenderse para:

- Integrales multidimensionales donde métodos deterministas fallan
- Técnicas avanzadas como muestreo por importancia
- Integración de funciones con singularidades
- Aplicaciones en física estadística y mecánica cuántica

La implementación proporciona una base sólida para estudios más avanzados en métodos de Monte Carlo y computación científica.