

# Comparación de Técnicas de Integración Numérica

Nicolás Gómez Aragón\*

Pontificia Universidad Javeriana

November 16, 2023

## Abstract

El propósito de este proyecto es llevar a cabo una evaluación y comparación de técnicas de integración numérica empleando funciones de prueba como referencia. Se investigarán métodos como la regla del trapecio, la regla de Simpson y la cuadratura gaussiana. Estas técnicas se implementarán en Python, y se llevará a cabo un análisis de sus resultados en términos de precisión y eficiencia computacional.

**Palabras Clave:** Integración, Eficiencia, computacional.

## Contents

<b>1</b>	<b>Resumen del Proyecto</b>	<b>2</b>
<b>2</b>	<b>Objetivos del Proyecto</b>	<b>2</b>
<b>3</b>	<b>Técnicas de Integración Numérica</b>	<b>2</b>
3.1	Regla del Trapecio . . . . .	2
3.2	Regla de Simpson . . . . .	2
3.3	Cuadratura Gaussiana . . . . .	3
<b>4</b>	<b>Funciones de Prueba</b>	<b>3</b>
4.1	Soluciones analíticas . . . . .	4
4.2	Reemplazo en intervalo . . . . .	4
<b>5</b>	<b>Implementación en Python</b>	<b>4</b>
<b>6</b>	<b>Resultados</b>	<b>5</b>
6.1	Eficiencia . . . . .	5
6.1.1	Función Trigonométrica . . . . .	5
6.1.2	Función Polinomial . . . . .	5
6.1.3	Función Racional . . . . .	5
6.2	Precisión . . . . .	5
6.2.1	Función Polinomial . . . . .	5
6.2.2	Función Racional . . . . .	5
6.2.3	Función Trigonométrica . . . . .	6
6.3	Análisis de resultados . . . . .	6
<b>7</b>	<b>Aplicaciones</b>	<b>6</b>
7.1	Problema de manejo de embalses . . . . .	6
7.2	OpenGeoSys . . . . .	6
7.3	Algoritmos en el Desarrollo de Motores de Juegos . . . . .	7
<b>8</b>	<b>Conclusiones</b>	<b>7</b>

---

\*nicolas.gomeza@javeriana.edu.co

# 1 Resumen del Proyecto

El objetivo de este proyecto es evaluar y comparar la precisión y eficiencia de varias técnicas de integración numérica utilizando funciones de prueba. Se explorarán métodos como la regla del trapecio, la regla de Simpson y la cuadratura gaussiana. Estas técnicas se implementarán en Python y se analizarán sus resultados en términos de precisión y eficiencia computacional.

## 2 Objetivos del Proyecto

- Profundizar en temas introductorios de análisis numérico y técnicas de integración numérica.
- Familiarizarse con implementaciones de métodos numéricos en Python para problemas de cálculo.
- Comprender conceptos clave como precisión y eficiencia computacional.

## 3 Técnicas de Integración Numérica

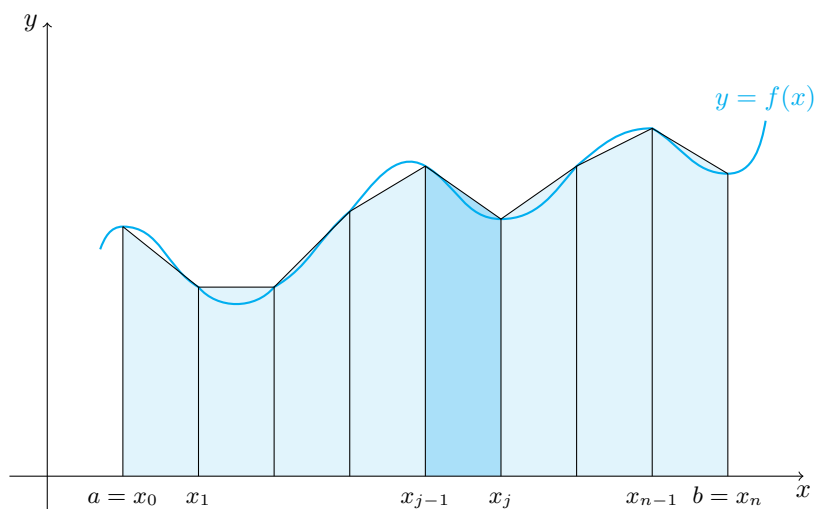
A continuación se describen las tres técnicas de integración las cuales serán estudiadas en este artículo.

### 3.1 Regla del Trapecio

La **Regla del Trapecio** es un método de aproximación numérica para calcular el valor de una integral definida. Se basa en dividir el área bajo una curva en múltiples trapecios y sumar sus áreas para estimar la integral. Para una función  $f(x)$  en el intervalo  $[a, b]$ , la fórmula de la Regla del Trapecio se expresa de la siguiente manera:

$$\int_a^b f(x)dx \approx \frac{b-a}{2} [f(a) + f(b)]$$

donde  $\frac{b-a}{2}$  es el ancho del trapecio y  $[f(a) + f(b)]$  representa la suma de las alturas de los extremos. Este método es simple, lo cual lo hace no muy preciso para funciones complicadas. [6]

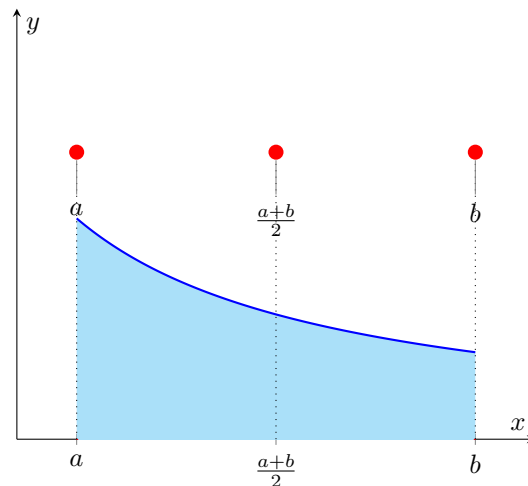


### 3.2 Regla de Simpson

La **Regla de Simpson** es un método más preciso que la Regla del Trapecio para la aproximación numérica de una integral definida. Utiliza polinomios de segundo grado para ajustar la curva de la función. La fórmula de la Regla de Simpson se expresa de la siguiente manera:

$$\int_a^b f(x)dx \approx \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

Este método utiliza tres puntos (los extremos  $a$  y  $b$  y el punto medio que es el promedio entre ambos) para ajustar una parábola, lo que lo hace más preciso que la Regla del Trapecio.[7]

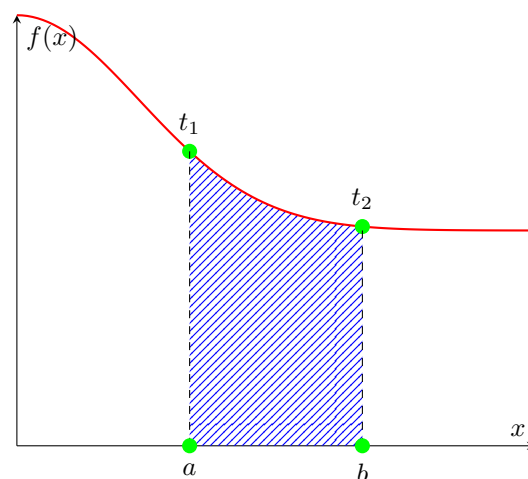


### 3.3 Cuadratura Gaussiana

La **Cuadratura Gaussiana** es un método de integración numérica que utiliza nodos y pesos específicos para aproximar la integral de una función en un intervalo  $[a, b]$ . La fórmula general de la Cuadratura Gaussiana se expresa de la siguiente manera:

$$\int_a^b f(x) dx \approx \sum_{i=1}^n w_i \cdot f(x_i)$$

donde  $w_i$  son los pesos asociados a los nodos  $x_i$ . La precisión de este método depende de la elección de los nodos y pesos, que varían según el tipo de Cuadratura Gaussiana utilizada (por ejemplo, Gauss-Legendre, Gauss-Hermite, Gauss-Laguerre, etc.).[3]



En la figura anterior se representa la aproximación de cuadratura gaussiana para la integración numérica de la función  $f(x) = e^{-x^2} + 1$  en el intervalo  $[1, 2]$ . Los puntos verdes,  $t_1$  y  $t_2$ , corresponden a los nodos de Gauss, donde se evalúa la función. Las líneas discontinuas verticales marcan las posiciones de estos nodos. La región sombreada ilustra el área bajo la curva de  $f(x)$  que se aproxima mediante polinomios que pasan por los puntos de Gauss, lo cual permite una estimación precisa de la integral de la función en el intervalo dado.

## 4 Funciones de Prueba

Para evaluar y probar los métodos de integración numérica, consideramos las siguientes tres funciones de prueba:

1. **Función Polinómica:** Esta función polinómica simple se puede integrar analíticamente, lo que la hace útil para comparaciones de precisión.

$$f(x) = x^2 + 3x - 5 \quad (1)$$

2. **Función racional:** Se optó por esta función por su simplicidad y la presencia de un término logarítmico en la integral, lo cual proporciona un buen ejemplo para evaluar la precisión de nuestras técnicas numéricas. Además, la integración de funciones racionales es una práctica común en análisis numérico y nos permite explorar cómo nuestros métodos se desempeñan en este contexto.

$$f(x) = \frac{1}{x+1} \quad (2)$$

3. **Función Trigonométrica:** La naturaleza trigonométrica de esta función nos permite probar cómo manejan los métodos numéricos el comportamiento oscilatorio.

$$f(x) = \sin(x) \quad (3)$$

## 4.1 Soluciones analíticas

Solucionamos analíticamente las funciones para encontrar las soluciones exactas a sus integrales para poder establecer un punto de comparación con las soluciones de los métodos de integración.

1. **Función polinómica:**

$$\int x^2 dx = \frac{1}{3}x^3 + C \quad (4)$$

Donde  $C$  es una constante de integración.

2. **Función racional:**

$$\int \frac{1}{x+1} dx = \ln|x+1| + C \quad (5)$$

3. **Función trigonométrica:**

$$\int \sin(x) dx = -\cos(x) + C \quad (6)$$

## 4.2 Reemplazo en intervalo

Tomando como punto de referencia un intervalo determinado para las tres funciones podemos comparar más fácilmente las soluciones analíticas con las funciones por métodos computacionales para su posterior análisis.

Así con esto, el intervalo será  $[0, 2]$  para las integrales definidas.

1. **Función polinómica:**

$$\int_0^2 x^2 dx = \frac{1}{3}x^3 \Big|_0^2 = \frac{1}{3}(2^3 - 0^3) = \frac{8}{3} = 2.666666667 \quad (7)$$

2. **Función racional:**

$$\int_0^2 \frac{1}{x+1} dx = \ln|x+1| \Big|_0^2 = \ln(3) - \ln(1) = \ln(3) = 1.098612289 \quad (8)$$

3. **Función trigonométrica:**

$$\int_0^\pi \sin(x) dx = -\cos(x) \Big|_0^\pi = -\cos(\pi) + \cos(0) = 2 \quad (9)$$

## 5 Implementación en Python

La implementaciones de cada técnica de integración numérica en el lenguaje de programación Python pueden encontrarse en el repositorio.

## 6 Resultados

### 6.1 Eficiencia

En este contexto eficiencia se refiere a la capacidad de un método o algoritmo de integración computacional para realizar cálculos de manera rápida y efectiva, minimizando el tiempo necesario para obtener resultados. La eficiencia es un aspecto crucial en la elección y aplicación de métodos de integración, ya que puede tener un impacto significativo en el rendimiento y la utilidad de las soluciones obtenidas.

En términos de eficiencia, medida como el tiempo de cálculo, se observaron los siguientes resultados:

#### 6.1.1 Función Trigonométrica

- Método más rápido: Trapezoidal (6.3e-05 segundos).
- Orden de eficiencia: Trapezoidal > Simpson > Gaussiano.

#### 6.1.2 Función Polinomial

- Método más rápido: Trapezoidal (2.1e-05 segundos).
- Orden de eficiencia: Trapezoidal > Simpson > Gaussiano.

#### 6.1.3 Función Racional

- Método más rápido: Trapezoidal (1.5e-05 segundos).
- Orden de eficiencia: Trapezoidal > Simpson > Gaussiano.

### 6.2 Precisión

Precisión en este contexto se refiere a la capacidad de un método computacional para obtener resultados que se acerquen de manera significativa a la solución analítica o teórica. La precisión es esencial en la evaluación de la confiabilidad de los resultados obtenidos mediante técnicas de integración, ya que nos permite determinar cuán cercanos están a las respuestas exactas.

En cuanto a la precisión, medida como la cercanía a la solución analítica, se obtuvieron los siguientes hallazgos:

#### 6.2.1 Función Polinomial

- Método más rápido: Trapezoidal con un tiempo de 1.2e-05 segundos.
- Seguido de: Simpson con una diferencia de 1.1130218505859372e-06 segundos respecto al método más rápido
- Seguido de: Gaussiana con una diferencia de 0.003229856491088867 segundos respecto al método más rápido
- Aproximación más cercana: Gaussiana.

#### 6.2.2 Función Racional

- Método más rápido: Trapezoidal con un tiempo de 1e-05 segundos.
- Seguido de: Simpson con una diferencia de 1.2056732177734367e-06 segundos respecto al método más rápido.
- Seguido de: Gaussiana con una diferencia de 0.00025463104248046875 segundos respecto al método más rápido.
- Aproximación más cercana: Gaussiana.

### 6.2.3 Función Trigonométrica

- Método más rápido: Trapezoidal con un tiempo de 4e-05 segundos.
- Seguido de: Simpson con una diferencia de 5.061111450195309e-06 segundos respecto al método más rápido.
- Seguido de: Gaussiana con una diferencia de 1.9073486328125e-06 segundos respecto al método más rápido.
- Aproximación más cercana: Gaussiana.

## 6.3 Análisis de resultados

Se observa una clara compensación entre eficiencia y precisión en los métodos evaluados. Mientras que los métodos Trapezoidal y Simpson son más rápidos, el método Gaussiano proporciona resultados más precisos.

La elección del método adecuado dependerá del equilibrio entre la necesidad de eficiencia (tiempo de cálculo) y precisión. Para funciones complejas o donde se requiere alta precisión, el tiempo adicional requerido por el método Gaussiano puede ser justificado, mientras que para respuestas rápidas con menor necesidad de precisión, los métodos Trapezoidal o Simpson pueden ser más adecuados.

## 7 Aplicaciones

En esta parte del proyecto se mostrarán 3 proyectos de aplicación donde los métodos computacionales, en especial los de integración juegan un rol determinante.

### 7.1 Problema de manejo de embalses

En el proyecto este proyecto de Universidad Autónoma de México (UNAM), se aborda una aplicación práctica de las técnicas de integración numérica en el ámbito de la ingeniería hidrológica. El estudio se enfoca en el uso de esquemas de integración numérica de Newton - la Regla del Trapecio, Simpson 1/3 y Simpson 3/8 - para calcular el volumen de almacenamiento en el embalse de la presa "Las Cruces", que forma parte de un proyecto hidroeléctrico. Este análisis resalta la relevancia de los métodos numéricos en la resolución de problemas complejos en el campo de la hidrología de superficie, una disciplina crucial en el diseño y manejo de infraestructuras hidráulicas.

Uno de los aspectos centrales del proyecto es la aplicación de la ecuación diferencial de continuidad para modelar el tránsito de avenidas en embalses, un enfoque que refleja la importancia de los métodos numéricos en el análisis hidrológico moderno. Además, se realiza una comparativa entre los volúmenes estimados mediante estos métodos de integración y los obtenidos a través de un programa de la Comisión que emplea una integración a través de rectángulos. Este contraste demuestra no solo la precisión de los métodos de Newton, sino también su utilidad práctica en escenarios de ingeniería real, proporcionando una base sólida para la toma de decisiones en proyectos de gran envergadura.[1]

### 7.2 OpenGeoSys

El proyecto OpenGeoSys (OGS) se presenta como una iniciativa científica de código abierto para la simulación numérica de procesos termo-hidro-mecánico-químicos (THMC) en medios porosos. Empleando principalmente el Método de Elementos Finitos (FEM), OGS proporciona un marco numérico flexible para abordar problemas multifacéticos con aplicaciones en geociencia e hidrología.

El modelado de procesos en OGS ha sido relevante en diversas áreas de la ingeniería y aplicaciones geocientíficas desde los 70s, abordando problemas como la consolidación del suelo, construcción de presas y exploración de campos de petróleo y gas. Los avances en estudios experimentales y teóricos sobre los efectos de temperatura, hidráulica, mecánica y química en medios porosos fracturados se han acelerado en las últimas dos décadas, impulsados por la necesidad de evaluar el rendimiento y la seguridad de los depósitos de residuos nucleares de alto nivel.

Para la evaluación del rendimiento y la seguridad a largo plazo de un repositorio de residuos nucleares en una formación geológica profunda, es crucial garantizar el aislamiento de un repositorio subterráneo. Se necesita predecir numéricamente los procesos de transporte de solutos bajo condiciones acopladas que involucren estabilidad mecánica, carga térmica de los residuos de alto nivel y química

en las aguas subterráneas. Con el rápido desarrollo de la tecnología informática, es posible analizar problemas geocientíficos complicados de manera acoplada utilizando códigos numéricos modernos. Sin embargo, entender estos procesos acoplados basados en datos experimentales disponibles e implementar los algoritmos desarrollados en los códigos numéricos sigue siendo un gran desafío para los científicos, requiriendo cooperación interdisciplinaria e interactiva.[4]

El sitio oficial del proyecto es página web y el código de este puede encontrarse en el siguiente repositorio.

### 7.3 Algoritmos en el Desarrollo de Motores de Juegos

Otra faceta de los métodos numéricos se encuentra el desarrollo de motores para videojuegos, incluyendo el Método de Euler, el Método de Verlet y el Método de Runge-Kutta para la integración numérica. La detección de colisiones y cómo se equilibra la simplicidad y precisión en este proceso, incluyendo el uso de volúmenes de límites geométricos y el algoritmo GJK son temas ampliamente estudiados, así como el cálculo de los puntos de contacto utilizando el Algoritmo de Sutherland-Hodgman para determinar la respuesta a colisiones.[5]

Un ejemplo de la importancia de los métodos de integración numérica y aprendizaje profundo en el desarrollo de videojuegos es el estudio realizado en el juego *DOOM*. En el artículo "Playing Doom with SLAM-Augmented Deep Reinforcement Learning", se explora un enfoque innovador para el aprendizaje de políticas en juegos 3D, utilizando un modelo de agente que incorpora conceptos semánticos y abstracciones. Este modelo se apoya en la Red de Aprendizaje Profundo Q (DQN) y mejora el procesamiento de entornos 3D complejos al añadir detalles sobre objetos y elementos estructurales, así como la localización del jugador.

El estudio demuestra cómo la introducción de mapas en el modelo DQN estándar, que proporcionan información agregada sobre el entorno 3D alrededor del personaje, mejora la eficacia del aprendizaje en *DOOM*. Estos mapas permiten al agente mantener información sobre áreas observadas anteriormente y construir un modelo agregado del entorno 3D, lo cual es crucial en el escenario de duelo a muerte de *DOOM*, donde el objetivo es eliminar tantos oponentes como sea posible. La investigación subraya los beneficios de las representaciones en el aprendizaje de políticas, ofreciendo perspectivas para mejorar el rendimiento y la escala de los algoritmos de aprendizaje en entornos 3D más allá de "DOOM".[2]

## 8 Conclusiones

Este proyecto ha explorado la utilidad y aplicación de algunos métodos de integración numérica, subrayando su importancia en una amplia gama de campos científicos. La integración numérica es fundamental en áreas donde las soluciones analíticas son difíciles o imposibles de obtener, como en la física computacional, la ingeniería, la economía, la modelización de fenómenos naturales y procesos industriales, entre otros.

A través del análisis de la Regla del Trapecio, la Regla de Simpson y la Cuadratura Gaussiana, hemos demostrado cómo cada técnica tiene sus propias ventajas y limitaciones, dependiendo del contexto. El método del Trapecio y la Regla de Simpson, debido a su mayor eficiencia en términos de tiempo de cálculo, son especialmente útiles en situaciones donde se requiere una respuesta rápida y la precisión absoluta no es crítica, mientras que la Cuadratura Gaussiana, con su mayor precisión, es invaluable en investigaciones científicas y aplicaciones técnicas donde la exactitud es primordial.

Este proyecto no solo ha proporcionado una comparación detallada de la eficiencia y precisión de estos métodos, sino que también ha brindado una plataforma para familiarizarse con su implementación en Python, así como también algunas de sus aplicaciones a nivel científico e industrial.

En conclusión, los métodos de integración numérica son herramientas poderosas y flexibles en el análisis numérico. Su comprensión y correcta aplicación son cruciales para resolver problemas complejos en el mundo moderno, y este proyecto ha planteado una introducción a su entendimiento, proporcionando familiaridad y un entorno para su aplicación en diferentes escenarios.

## References

- [1] Maritza Liliana Arganis-Juárez et al. "Newton's numerical integration schemes applied in a reservoir operation problem". In: *Volumen XIX (número 2), abril-junio 2018* (2018). Artículo arbitrado,

- pp. 183–193. ISSN: 2594-0732. DOI: 10.22201/fi.25940732e.2018.19n2.016. URL: <http://dx.doi.org/10.22201/fi.25940732e.2018.19n2.016>.
- [2] Shehroze Bhatti et al. *Playing Doom with SLAM-Augmented Deep Reinforcement Learning*. 2016. arXiv: 1612.00380 [cs.AI].
  - [3] Víctor Elvira, Luca Martino, and Pau Closas. “Importance Gaussian Quadrature”. In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 474–488. DOI: 10.1109/TSP.2020.3045526.
  - [4] Olaf Kolditz et al. “OpenGeoSys: An open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/C) processes in porous media”. In: *Environmental Earth Sciences* 67 (Sept. 2012), pp. 589–599. DOI: 10.1007/s12665-012-1546-x.
  - [5] Harold Serrano. *Algorithms in Game Engine Development*. For a game character to experience physics, a game engine needs to compute the equations of motion, collision, contact points, etc. There is a set of basic algorithms that aids a character experience such effects. For example, the Runge-Kutta Method computes the Equations of Motion using numerical integration methods. The Gilbert-Johnson-Keerthi (GJK) algorithm detects collision using the Minkowski Difference. The Sutherland-Hodgman algorithm identifies collision contact points by clipping a polygon. 2016. URL: <https://www.haroldserrano.com/blog/algorithms-in-game-engine-development> (visited on 11/16/2023).
  - [6] G. K. Smyth. “Numerical Integration”. In: *Encyclopedia of Biostatistics*. Ed. by P. Armitage and T. Colton. London: Wiley, 1998, pp. 3088–3095. URL: <https://gksmyth.github.io/pubs/NumericalIntegration-Preprint.pdf>.
  - [7] R. B. Srivastava and Jai Singh Yadav. “Relative Accuracy of Numerical Quadrature Formulas - Trapezoid Rule, Simpson’s 1/3 Rule, Simpson’s 3/8 Rule and Boole’s Rule”. In: *Recent Research in Science and Technology* 3.7 (2011), pp. 77–84. ISSN: 2076-5061. URL: <https://core.ac.uk/download/pdf/236009152.pdf>.