Diferenciación e Integración

Juan Manuel Cuellar Borrero

Pontificia Universidad Javeriana Cali

Cali, Colombia

Juanma1909@javerianacali.edu.co

Abstract – En el presente documento se explicarán los procesos llevados a cabo para resolver el problema de diferenciación e integración.

I.INTRODUCCION

Se desarrollará un archivo utilizando el lenguaje de programación Python, cumpliendo con los requisitos necesarios para dar solución al problema de diferenciación e integración, a través de la implementación de los algoritmos de Diferencias finitas hacia adelante, Diferencias finitas hacia atrás, Diferencias centradas, estos para la diferenciación, y para integración los métodos de cuadratura compuesta de rectángulo, trapezoide y Simpson.

II.DESARROLLO DE CONTENIDO

1. ***Materiales y Métodos***
1. **Materiales**

- **Python:** Este fue el lenguaje utilizado para desarrollar los algoritmos de Diferencias finitas hacia adelante, Diferencias finitas hacia atrás, Diferencias centradas, estos para la diferenciación, y para integración los métodos de cuadratura compuesta de rectángulo, trapezoide y Simpson "Python es un lenguaje de programación de código abierto, orientado a objetos, muy simple y fácil de entender. Tiene una sintaxis sencilla que cuenta con una vasta biblioteca de herramientas, que hacen de Python un lenguaje de programación único." [1].

- **Librería Numpy:** Esta librería ayudo a poder realizar todas las operaciones relacionadas con algebra lineal de forma rápida y efectiva. " NumPy es un paquete de Python que significa "Numerical Python", es la librería principal para la informática científica, proporciona potentes estructuras de datos, implementando matrices y matrices multidimensionales. Estas estructuras de datos garantizan cálculos eficientes con matrices. " [2].

- **Librería Matplotlib:** Esta es la librería utilizada para poder graficar tanto los polinomios de ajuste como los puntos utilizados de las bases de datos. "Matplotlib es una librería de trazado utilizada para gráficos 2D en lenguaje de programación Python, es muy flexible y tiene muchos valores predeterminados incorporados que te ayudarán muchísimo en tu trabajo." [3].

- **Librería Time:** Esta librería ayuda a tomar los tiempos de ejecución de los algoritmos.

2. **Metodología Diferenciación**

Para la elaboración del algoritmo de Diferencias finitas hacia adelante, se hizo uso de la formula $f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 \ldots$ En la que al despejar la primera derivada y mantenerlo en una precisión de orden 1 quedaría $f'(x) = \frac{f(x+h)-f(x)}{h}$.

Por otro lado, para el algoritmo de diferencias finitas hacia atrás se utiliza una fórmula similar, siendo esta $f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2}h^2$.. De igual manera, al despejar la primera derivada y mantenerlo en una precisión de orden 1 quedaría $f'(x) = \frac{f(x)-f(x-h)}{h}$.

Por último, para las diferencias centradas, se le resta a la primera serie la segunda serie, el resultado al despejar la primera derivada se obtiene $f'(x) = \frac{f(x+h)-f(x-h)}{2h}$ la cual tiene precisión de orden 2.

Para realizar pruebas se utilizaron 3 funciones distintas, estas fueron $\ln(x)$, $Sin(2x)$ y e^{2x}.

1. ***Resultados utilizando la función trigonométrica:***

h	Atras		Adelante		Centrada	
1	1.132	0.549	1.146	0.550	0.692	0.334
0.1	0.127	0.061	0.127	0.061	0.008	0.004
0.01	0.012	0.006	0.012	0.006	8x 10^{-5}	4x10^{-5}
0.001	0.001	6x 10^{-4}	0.001	6x 10^{-4}	8x 10^{-7}	4x10^{-7}

Tabla 1. Error Medio y desviación estándar por método

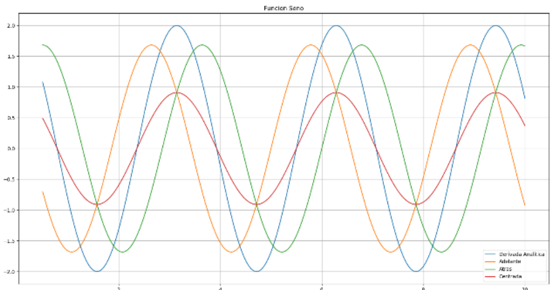

Figura 1. Función seno con h = 1

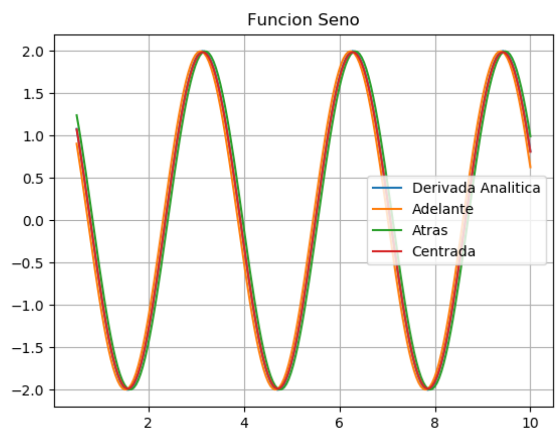

Figura 2. Función seno con h = 0.1

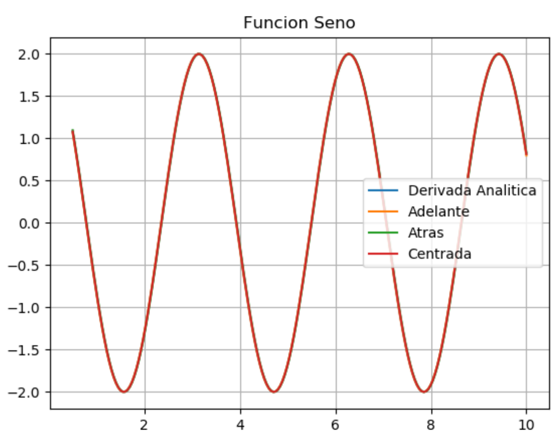

Figura 3. Función seno con h = 0.01

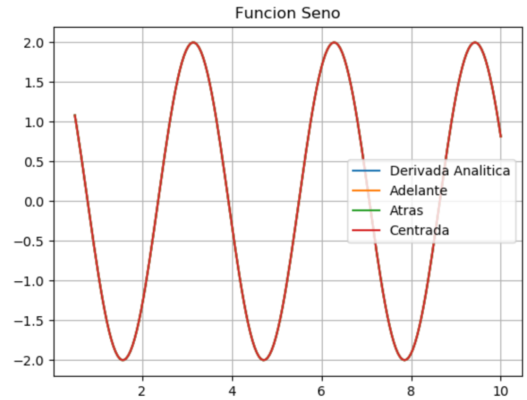

Figura 4. Función seno con h = 0.001

2. **Análisis**:
   A primera vista, es fácil notar el hecho de que a medida que el h se hace más pequeño, las gráficas se ajustan más al valor dado por la derivada analítica, esto tiene sentido pues en la teoría de diferenciación, se realiza un límite cuando h tiende a 0. Además, cabe resaltar que el método de diferencias centrada, siempre tuvo tanto un error medio como una desviación estándar, mucho menor a sus contrapartes.

2. ***Resultados utilizando la función logarítmica***

h	Atras		Adelante		Centrada	
1	0.083	0.190	0.035	0.048	0.023	0.073
0.1	0.004	0.007	0.004	0.006	1x 10^{-4}	3x 10^{-4}
0.01	4x 10^{-4}	7x 10^{-4}	4x 10^{-4}	7x 10^{-4}	1x 10^{-6}	3x 10^{-5}
0.001	4x 10^{-5}	7x 10^{-5}	4x 10^{-5}	7x 10^{-5}	1x 10^{-8}	3x 10^{-8}

Tabla 2. Error medio y desviación estándar por método

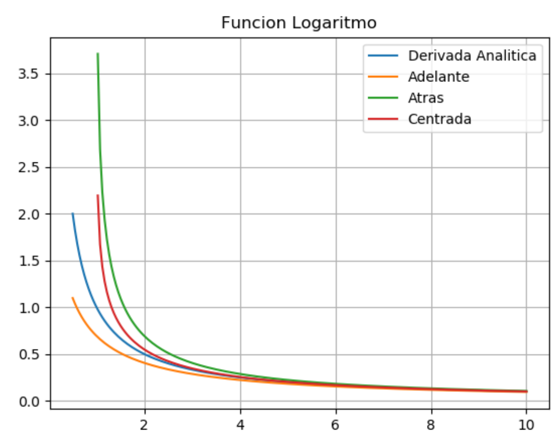

Figura 5. Función logaritmo natural con h = 1

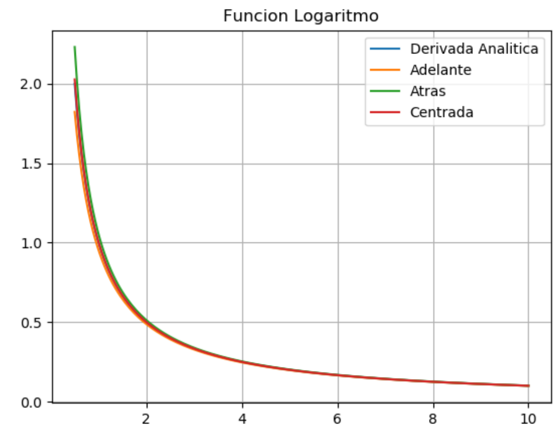

Figura 6. Función logaritmo natural con h = 0.1

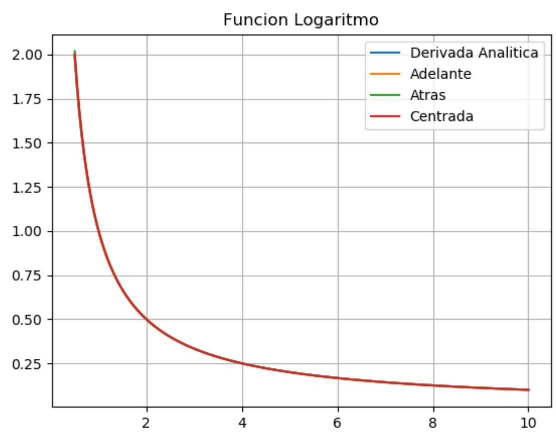

Figura 7 Función Logaritmo natural con h = 0.01

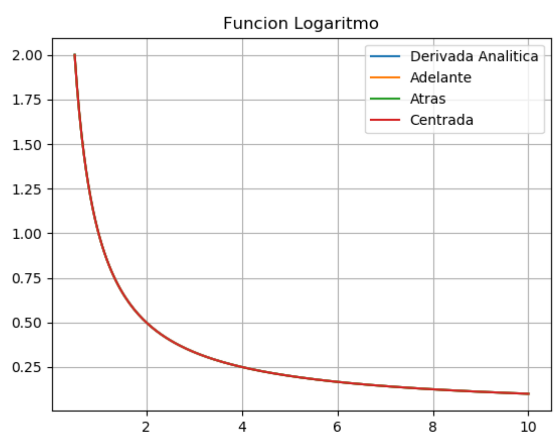

Figura 8. Función Logaritmo natural con h = 0.001

3. **Analisis:**

Nuevamente se ve que a medida que el valor de h se aproxima a 0, el error medio y desviación estándar de todas las metodologías disminuye, al igual que con la función anterior, la metodología de diferencia centrada es la que más se acerca al valor real. Otra observación es que tanto los errores medios como las desviaciones estándar son menores que los obtenidos con la función trigonométrica, lo que indica que la precisión también depende de la función a diferenciar.

3. ***Resultados usando la función e^2x***

h	Atras		Adelante		Centrada	
1	302x 10^4	883 x 10^4	116x 10^5	341x 10^5	433x 10^4	126x 10^5
0.1	498x 10^3	145 x 10^4	570x 10^3	166x 10^4	355x 10^2	104x 10^3
0.01	529x 10^2	154 x 10^3	536x 10^2	156x 10^3	3552	103x 10^2
0.001	5324	155 x 10^2	5331	155x 10^2	35	103

Tabla 3. Error medio y desviación estándar por método

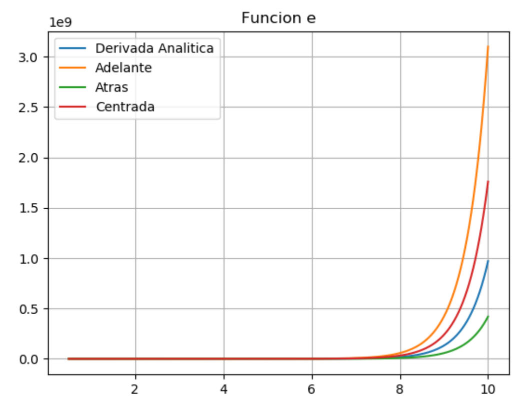

Figura 9. Funcion e^2x con h = 1

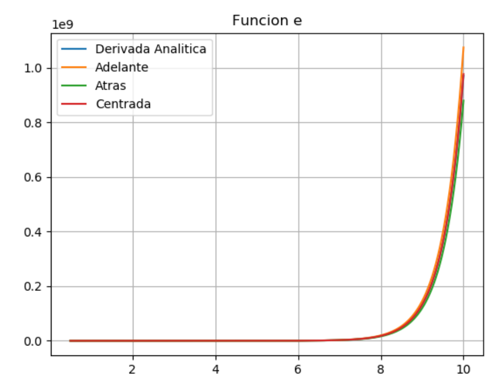

Figura 10 Función e^2x con h = 0.1

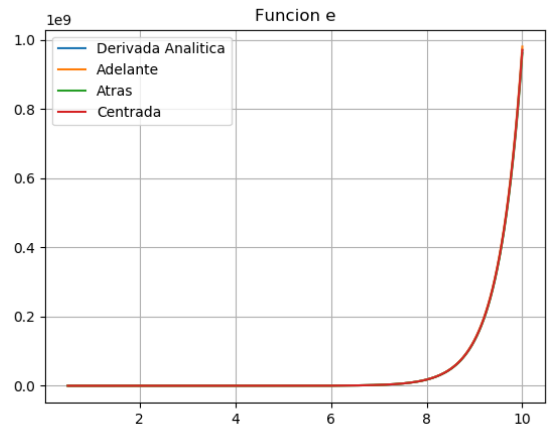

Figura 11. Función e^2x con h = 0.01

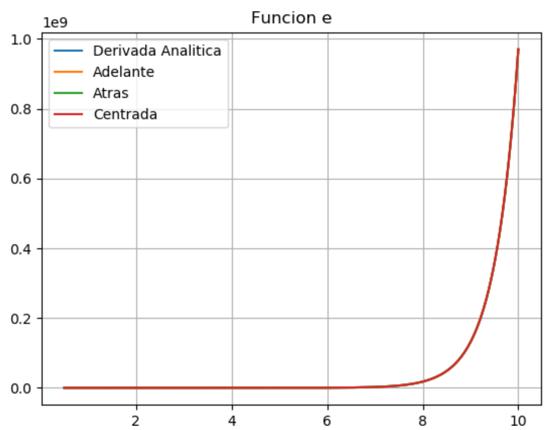

Figura 12. Función e^2x con h = 0.001

4. **Analisis:**
En primer lugar, se puede notar que los errores con respecto a esta función son demasiado altos, se cree que esto es debido a que los valores que toma la función en el eje "y", llegan a estar en la escala de 10^9 tal y como muestran las gráficas. Además de esto se ve que conforme h se acerca a 0 el error disminuye de una forma bastante considerable.

- **Aclaraciones:** Para la interpretación de las tablas mostradas en el documento, se debe tomar en cuenta que para cada método (Atrás, Adelante y Centrado) el valor que aparece de primero hace referencia al error medio, con el h en cuestión, y el segundo valor hace referencia a la desviación estándar.

- **Metodología Integracion**
Para realizar la aproximación de las integrales se realizaron los algoritmos de cuadratura compuesta para los métodos de Rectángulo, Trapezoide y Simpson. En esta lo que se hace es subdividir el intervalo de integración [a,b] en n subintervalos, posteriormente para cada subintervalo se aplica el metodo de aproximación de integración a utilizar y se va acumulando los resultados de cada subintervalo, al final

Resultados para a=1 , b=4 y n = 10:

Funcion	Rectangulo	Trapexoide	Simpson
Sen(2x)	0.268	0.262	0.266
Ln(x)	0.429	0.421	0.426
e^2x	1185	1306	1225

Tabla 4. Errores de aproximación de integrales

3. **Analisis:**
Es fácil notar que la precisión de cada método depende mucho de la clase de función a la que se le pretende hallar la integral.

Resultados para a =1 , b = 10 y n = 15

Funcion	Rectangulo	Trapexoide	Simpson
Sen(2x)	0.564	0.537	0.555
Ln(x)	1.412	1.372	1.399
e^2x	$5164x10^4$	$6572x10^4$	$5633x10^4$

4. **Analisis:**
Nuevamente aparece el problema que la función e^(2x) arroja valores de escala 10^9, esto hace que el error cuando el límite de integración superior sea relativamente alto.

Resultados para a = 2, b = 5 y n = 40

Funcion	Rectangulo	Trapexoide	Simpson
Sen(2x)	8.697	$1x10^{-4}$	1.630
Ln(x)	7.030	$1x10^{-4}$	2.569
e^2x	10.292	20.590	0.0019

5. **Analisis**
Aquí se ve como el aumentar el número de subintervalos, puede llegar a bajar el error con respecto a la inegral deseada, siendo mejor utilizar n con valores más altos.

III. CONCLUSIONES

En primer lugar, y lo que más llama la atención es el hecho de que la precisión de los algoritmos de diferenciación depende mucho de la función a la que se le pretende obtener la derivada. Además de ello se ve claramente como el tomar valores de h más cercanos a 0, disminuye notablemente el error de los algoritmos con respecto a la derivada analítica, por lo que se recomienda tomar valores muy cercanos a 0 para poder obtener mejores resultados. Por último, el método de diferencias centradas siempre obtuvo los valores de error más pequeño con respecto a sus otras 2 contrapartes, lo que lo hace el método predilecto.

Para la aproximación de integrales, se puede ver que depende mucho no solo de la función a la que se le piensa hallar el área bajo la curva, sino también del intervalo de integración que se quiere evaluar, pues ciertas funciones pueden comportarse de una manera en cierto intervalo Q, y en dicho intervalo Q un método en específico obtiene mejores resultados, mientras que en el resto de intervalos otros métodos podrían acomodarse mejor.

Con respecto a los tiempos de ejecución de los algoritmos de integración y diferenciación, se decidió no reportarlo puesto que estos variaban demasiado, tanto en integración como en diferenciación, es decir, para los mismos valores de entrada, es una corrida arrojaba tiempos de 0.0 seg y en otros 0.00015(aprox)seg, el por ello que se puede decir con bastante seguridad estos algoritmos no tienen una complejidad temporal muy alta, pues ni siquiera con la librea time de Python se lograba obtener tiempos certeros. Esto debido a que lo que se realizan son operaciones matemáticas sencillas que se pueden calcular en O(1)

IV. REFERENCIAS

[1] A. Soloaga (2018, Octubre 19) “Principales Usos dePython”.[Online]. Disponible en : https://www.akademus.es/blog/programacion/principales-usos-python/

[2]. L. González. (2018, Septiembre 21) “Introducción a la librería NumPy de Python – Parte 1 ``.[Online]. Disponible en : https://ligdigonzalez.com/introduccion-a-numpy-python-1/

[3]. L. González. (2018, Octubre 19). “Introducción a la Librería Matplotlib de Python – Parte 1 ``.[Online]. Disponible en : https://ligdigonzalez.com/libreria-pandas-de-matplotlib-tutorial/