



Universidad
de La Laguna

Método de Simpson

Melanie Hernández, Nadia Chinaa, Tania Gutiérrez

17 de mayo de 2013

Facultad de Matemáticas
Universidad de La Laguna

Índice

1 Motivación y Objetivos

Índice

- 1 Motivación y Objetivos
- 2 Fundamentos Teóricos

Índice

- 1 Motivación y Objetivos
- 2 Fundamentos Teóricos
- 3 Procedimiento experimental
 - Descripción de los experimentos
 - Descripción del material
 - Resultados obtenidos
 - Análisis de los resultados

Índice

- ➊ Motivación y Objetivos
- ➋ Fundamentos Teóricos
- ➌ Procedimiento experimental
 - Descripción de los experimentos
 - Descripción del material
 - Resultados obtenidos
 - Análisis de los resultados
- ➍ Conclusiones

Objetivo principal

La intención principal es el planteamiento de un experimento cuyo fin es el cálculo del área de una función dada de la forma más precisa posible. Interviene un método de integración numérica, por lo tanto conlleva no sólo un aprendizaje en el ámbito informático sino que además permite hacer un enfoque matemático que obliga a alcanzar una mayor destreza en este campo.

Objetivos

- Uso del \LaTeX y comprensión de todas sus sentencias de una forma ordenada y precisa como se muestra en este informe.
- Utilización de un macro de \TeX , Beamer, para desarrollar capacidades de relación y exposición de un tema mediante transparencias con la intención de transmitir y mostrar de una forma simple lo que hemos aprendido a traducir a \LaTeX .
- Manejo de Python, lenguaje principal que permitirá la realización del experimento y la obtención de datos.

Fundamentos Teóricos

La forma más común a la hora de hallar áreas de funciones es la integral definida. La integración es un concepto fundamental del cálculo y del análisis matemático. Básicamente, la integral es una generalización de la suma de infinitos sumandos, infinitamente pequeños.

Algunos métodos numéricos son el método de Taylor, Simpson, punto medio, regla del rectángulo, etc. En este caso, utilizaremos el método de Simpson.

$$\int_{-1}^1 \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} dx$$

Para la regla de Simpson existen dos fórmulas cuya elección recae en el tamaño de la acotación, es decir, para un intervalo pequeño, como por ejemplo $[-1,1]$, se debe utilizar la fórmula simple del método y para una cota superior, como lo es $[2,23]$ es más correcto el uso de la fórmula compuesta de Simpson.

Fórmulas

La fórmula simple se define como:

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

Por otra parte, la fórmula compuesta aparece como:

$$\int_a^b f(x)dx \approx \frac{h}{3} \left[f(x_0) + 2 \sum_{j=1}^n \frac{1}{2} f(x_{2j}) + 4 \sum_{j=1}^n \frac{1}{2} f(x_{2j-1}) + f(x_n) \right]$$

Procedimiento experimental

El experimento se basará en la comparación entre las fórmulas, en el que se realizarán pruebas de tiempo de ejecución de ambas con diferentes intervalos, además de distintas reiteraciones en el caso de la compuesta. Por lo tanto se puede dividir en dos partes.

Esta parte del experimento comienza con la creación de un módulo principal que contiene una definición de la función y llamadas al usuario con la intención de que introduzca los intervalos deseados. Los resultados que se obtienen se guardarán en un fichero que se pedirá también por teclado, pudiendo guardar varias ejecuciones en el mismo, es decir, sin que se sobrescriba al usar el mismo nombre. En un módulo aparte se calculará el tiempo que tarda en ejecutarse cada intervalo y el tiempo que mantiene el cpu.

Ejemplo

- *Con el intervalo $[-1,1]$*
- *Con el intervalo $[-1,4]$*
- *Con el intervalo $[2,8]$*

En cuanto a la segunda parte, se podría decir que es una ampliación de la primera. Dado que la fórmula compuesta permite realizar el experimento con números de reiteraciones diferentes. Se proponen dos programas en los que el número de reiteraciones esté ya asignado, siendo 2 y 4 el número de repeticiones que se harán. Al igual que en la primera parte verán los tiempos transcurridos y los del CPU.

Hardware y Software

Ejemplo

- 1 *Tipo de CPU: Pentium(R) Dual-Core CPU E5200 @ 2.50GHz*

Hardware y Software

Ejemplo

- 1 *Tipo de CPU: Pentium(R) Dual-Core CPU E5200 @ 2.50GHz*
- 2 *cache size: 2048 KB*

Hardware y Software

Ejemplo

- 1 *Tipo de CPU: Pentium(R) Dual-Core CPU E5200 @ 2.50GHz*
- 2 *cache size: 2048 KB*
- 3 *CPU speed: 1200.000 Hz*

Hardware y Software

Ejemplo

- 1 *Tipo de CPU: Pentium(R) Dual-Core CPU E5200 @ 2.50GHz*
- 2 *cache size: 2048 KB*
- 3 *CPU speed: 1200.000 Hz*
- 4 *vendor ID: GenuineIntel*

Hardware y Software

Ejemplo

- 1 *Tipo de CPU: Pentium(R) Dual-Core CPU E5200 @ 2.50GHz*
- 2 *cache size: 2048 KB*
- 3 *CPU speed: 1200.000 Hz*
- 4 *vendor ID: GenuineIntel*

Medidas de tiempo y Velocidad

Intervalo	Area
$[-1,1]$	0693236856881
$[-1,4]$	0.633179114507
$[2,8]$	0.0539969133913

Cuadro: Resultados experimentales del calculo del area (formula simple)

Intervalo	Area con iteracion = 4
$[-1,1]$	0.5813286
$[-1,4]$	0.646540956
$[2,8]$	0.0278711396576

Cuadro: Resultados experimentales del calculo del area (formula compuesta)

Intervalo	Tiempo transcurrido	Tiempo CPU
$[-1,1]$	$1.096754 \text{ e}^{-0,5}$	0.0
$[-1,4]$	$1.2159347\text{e}^{-0,5}$	0.0
$[2,8]$	$1.096725 \text{ e}^{-0,5}$	0.0

Cuadro: Resultados experimentales de tiempo(formula simple)

Intervalo	Tiempo transcurrido	Tiempo CPU
$[-1,1]$	$9.059906e^{-0,5}$	0.0
$[-1,4]$	$1.001358 e^{-0,5}$	0.0
$[2,8]$	$7.867813 e^{-0,6}$	0.0

Cuadro: Resultados experimentales de tiempo (formula compuesta)

Representaciones de las funciones

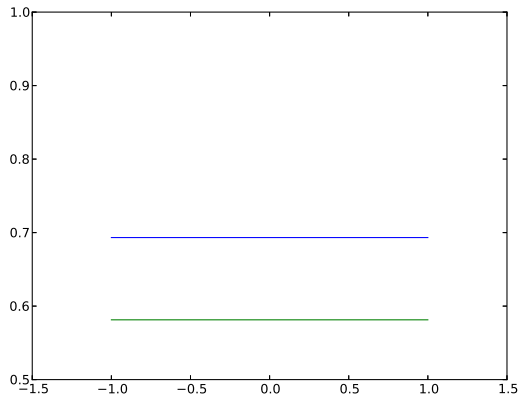


Figura: Intervalo $[-1,1]$

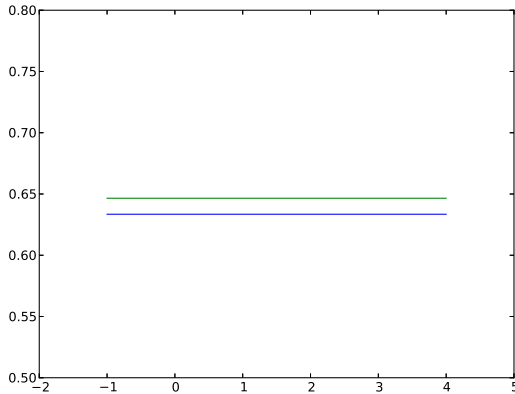


Figura: Intervalo $[-1,4]$

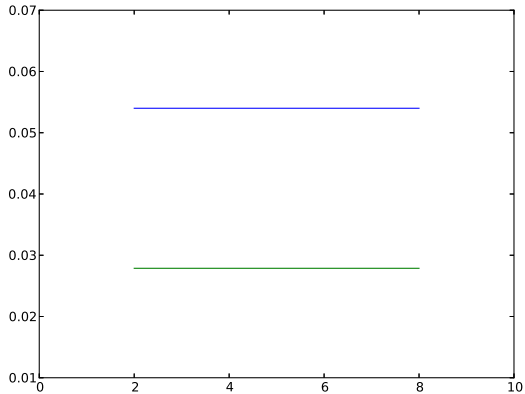


Figura: Intervalo $[2,8]$

De las gráficas podemos decir que con los intervalos $[-1,1]$ y $[2,8]$, las rectas que se forman están más alejadas entre sí que en el intervalo que podríamos considerar ni grande ni pequeño. Se ha experimentado con el tiempo transcurrido para el tratamiento del método y con el de ejecución de la CPU. Como sabemos, el tiempo transcurrido durante la ejecución del programa varía dependiendo de la computadora que se use, aún así podemos apreciar que es mayor al realizar la fórmula compuesta debido a que las reiteraciones hacen que el programa se retase. En cuanto al tiempo de la CPU, no varía en absoluto en ninguno de los intervalos porque la respuesta de la CPU es inmediata.

Conclusiones

El objetivo de este trabajo ha sido la confección de un informe redactado en código $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, y unas transparencias en Beamer con este mismo lenguaje. Cabe decir, que el lenguaje de programación con el que se ha confeccionado el experimento ha sido el python. Hemos obtenido la verificación de que los intervalos menores se usan para la fórmula simple y que los mayores para la fórmula compuesta. También hemos visto que en los programas en los que intervienen definiciones de funciones no muy complejas el tiempo de la CPU es mínimo. Se puede concluir diciendo que hemos obtenido facilidades a la hora de creación de códigos para resolución de problemas, además de haber logrado aumentar las capacidades que ya se tenían con $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$