



### **Grupo 3**

Miguel Angel Avila Santos  
Juan Andres Martinez Amado  
Jorge Luis Esposito Albornoz  
Juan Sebastian Herrera Guaitero

### **Asignatura**

Análisis numérico

### **Profesora**

Eddy Herrera Daza

### **Taller 4**

Integración y Ecuaciones Diferenciales

4 de Noviembre de 2021

## 1. Integración

1. c. Dados los siguientes puntos:

(0.1,1.8), (0.2,2.6),(0.3,3.0),(0.4,2.8),(0.5,1.9)

Utilice la fórmula de Simpson para encontrar una aproximación del área bajo la curva y calcule su error de truncamiento. Qué resultado se obtendría si primero interpola con Lagrange y luego calcula la integral compare los resultados con respecto al área.

### Código:

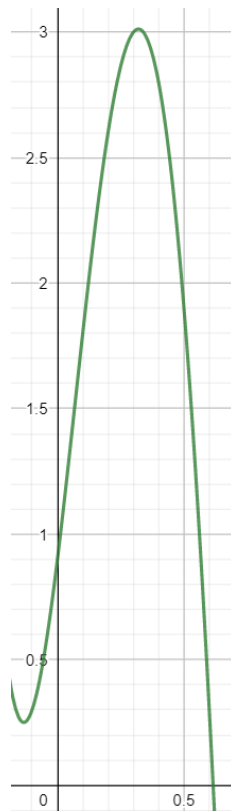
```
from scipy import integrate, interpolate

x = (0.1, 0.2, 0.3, 0.4, 0.5)
y = (1.8, 2.6, 3.0, 2.8, 1.9)

simpson_result = integrate.simpson(y, x)
print("Area bajo la curva(simpson):", simpson_result)
ec_lagrange_interpolation = interpolate.lagrange(x, y)
res_lagrange = integrate.simpson(ec_lagrange_interpolation(x),
x)
print('-----')
print('|      POLINOMIO INTERPOLADO(LAGRANGE)      |')
print('-----')
print(ec_lagrange_interpolation)
print('-----')
print("Area bajo la curva(simpson) con la interpolacion lagrange:",
res_lagrange)
error = simpson_result - res_lagrange
print("Error Truncamiento:", error)
```

### Resultados:

```
Area bajo la curva(simpson): 1.0433333333333334
-----
|      POLINOMIO INTERPOLADO(LAGRANGE)      |
-----
      4      3      2
41.67 x - 75 x + 14.58 x + 8.25 x + 0.9
-----
Area bajo la curva(simpson) con la interpolacion lagrange: 1.0433333333333057
Error Truncamiento: 2.7755575615628914e-14
```



Gráfica con la interpolación de Lagrange