Juan Felipe Arias

Natalia Gaona

Laura Sofia Jiménez

Informe:

Ejercicio:

Integración

a. **G1:** Teniendo en cuenta que en la regla de los trapecios el error de truncamiento está dado por:

$$T = -\frac{h^2}{12}(b-a)\,f''(z), \quad a \le z \le b$$

Estime el número mínimo de trapecios para aproximar $\int_0^2 \sin 2x dx$ donde la respuesta tenga un error absoluto menor de 0.0001.

b. G2: Aplique lo anterior (numeral a), para aproximar $\int_0^2 \sqrt{x} sinx dx$ y evaluar el error

Regla del trapecio:

La regla del trapecio es un método de integración numérica en donde se calcula aproximadamente el valor de una integral definida, se basa en aproximar el valor de la integral f(x) por el de la función lineal que pasa a través de (a,b). La integral es igual al área del trapecio bajo la gráfica de la función lineal. Se define de la siguiente manera:

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

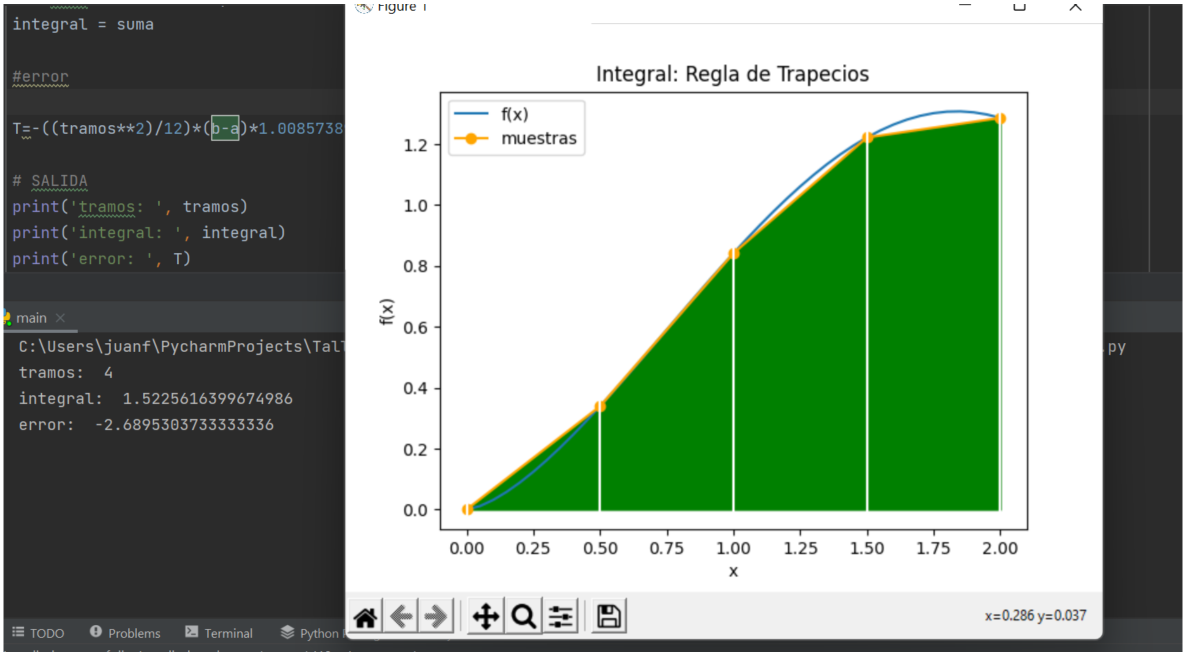
Código:

```python
# Integración: Regla de los trapecios
# Usando incluso muestras arbitrariamente espaciadas
import numpy as np
import matplotlib.pyplot as plt

# INGRESO
fx = lambda x: np.sqrt(x)* np.sin(x)

# intervalo de integración
a = 0
b = 2
tramos = 4

# PROCEDIMIENTO
# Puntos de muestra
muestras = tramos + 1
xi = np.linspace(a,b,muestras)
fi = fx(xi)
```

```python
# Regla del Trapecio
# Usando puntos muestreados
# incluso arbitrariamente espaciados
suma = 0
for i in range(0,tramos,1):
    dx = xi[i+1]-xi[i]
    Atrapecio = dx*(fi[i]+fi[i+1])/2
    suma = suma + Atrapecio
integral = suma

# SALIDA
print('tramos: ', tramos)
print('integral: ', integral)

# GRAFICA
# Puntos de muestra
muestras = tramos + 1
xi = np.linspace(a,b,muestras)
fi = fx(xi)
```

```python
muestraslinea = tramos*10 + 1
xk = np.linspace(a,b,muestraslinea)
fk = fx(xk)

# Graficando
plt.plot(xk,fk, label ='f(x)')
plt.plot(xi,fi, marker='o',
         color='orange', label ='muestras')

plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Integral: Regla de Trapecios')
plt.legend()

# Trapecios
plt.fill_between(xi,0,fi, color='g')
for i in range(0,muestras,1):
    plt.axvline(xi[i], color='w')

plt.show()
```

Resultados:



Conclusión:

La integración con métodos numéricos es una herramienta útil cuando se integra una función muy complicada, de acuerdo a los resultados podemos darnos cuenta que la interpolación lineal es precisa y sencilla para la determinación del área bajo la curva, pero para esto los intervalos deben ser mas pequeños.