

# INFORME LAB 04 – FISICA COMPUTACIONAL

**Link código:**

<https://drive.google.com/file/d/1jKn2SXmwNCWOV7zHZAjrFgiK63GGolQp/view?usp=sharing>

## Introducción

El trabajo realizado por una fuerza sobre un cuerpo puede calcularse utilizando dos enfoques: la integral y la sumatoria. Este informe presenta un análisis comparativo entre ambos métodos para determinar el trabajo en diferentes funciones de fuerza (polinómica, seno, coseno y exponencial). El objetivo es evaluar la precisión de la sumatoria en relación a la integral y calcular la diferencia porcentual entre ambos métodos.

## Marco Teórico

El trabajo mecánico se define como el producto de una fuerza aplicada sobre un objeto y la distancia recorrida en la dirección de la fuerza. Matemáticamente, se expresa como una integral definida:

$$W = \int_a^b F(x) dx$$

La integral permite calcular el área bajo la curva de la función de fuerza, mientras que la sumatoria aproxima dicha área dividiendo el intervalo en pequeñas secciones y sumando el área de cada segmento. La diferencia porcentual entre ambos métodos se usa para medir la precisión de la aproximación:

$$C\% = \left| \frac{\text{Integral} - \text{Sumatoria}}{\text{Integral}} \right| \times 100$$

## Desarrollo

El desarrollo del código incluye varios componentes que permiten calcular y graficar el trabajo utilizando la integral y la sumatoria para diferentes funciones de fuerza:

### 1. Definición de las funciones de fuerza:

- Se definen cuatro funciones de fuerza diferentes: polinómica, seno, coseno y exponencial. Cada una de estas funciones está implementada como una función en Python que toma un valor de entrada  $x$  y devuelve el valor de la fuerza en ese punto.

```
def fuerza_polinomica(x):
    return 2*x**2 + 3*x + 1
```

## 2. Selección de la función de fuerza:

- El usuario puede seleccionar la función de fuerza que desea utilizar mediante un menú interactivo. La selección se realiza a través de un menú de opciones, donde cada número corresponde a una función específica. El código valida la opción ingresada y asigna la función y el título correspondiente.

```
# Selección de la función de fuerza
def seleccionar_funcion():
    print("Seleccione la función de fuerza:")
    print("1: Polinómica (2*x^2 + 3*x + 1)")
    print("2: Seno (sin(x))")
    print("3: Coseno (cos(x))")
    print("4: Exponencial (exp(x))")
    opcion = int(input("Ingrese el número de su opción: "))

    if opcion == 1:
        return fuerza_polinomica, "Fuerza Polinómica"
    elif opcion == 2:
        return fuerza_seno, "Fuerza Senoidal"
    elif opcion == 3:
        return fuerza_coseno, "Fuerza Cosenoidal"
    elif opcion == 4:
        return fuerza_exp, "Fuerza Exponencial"
    else:
        print("Opción no válida.")
        exit()
```

## 3. Cálculo del trabajo mediante la integral:

- Se utiliza la función `quad` de la librería `scipy.integrate` para calcular la integral definida de la función de fuerza en el intervalo de 0 a 1. La función devuelve el valor del trabajo calculado de manera exacta.

```
# Calcular el trabajo usando la integral
trabajo_integral, _ = quad(fuerza, a, b)
```

## 4. Cálculo del trabajo mediante la sumatoria:

- La sumatoria se realiza dividiendo el intervalo de integración en 1000 subdivisiones. Se calcula el área de cada segmento multiplicando el valor de la función en cada punto por el ancho del segmento ( $dx$ ). La sumatoria de estos valores proporciona una aproximación del trabajo.

```
# Calcular el trabajo usando la sumatoria
x = np.linspace(a, b, n)
dx = (b - a) / n
trabajo_sumatoria = np.sum(fuerza(x) * dx)
```

## 5. Cálculo de la diferencia porcentual:

- La diferencia porcentual entre el valor calculado por la integral y la sumatoria se calcula utilizando la fórmula especificada anteriormente. Este valor permite evaluar la precisión de la sumatoria en comparación con la integral exacta.

```
# Calcular la diferencia porcentual
diferencia_porcentual = abs((trabajo_integral - trabajo_sumatoria) / trabajo_integral) * 100
```

## 6. Gráfica de resultados:

- Se utiliza matplotlib para graficar la función de fuerza, el área bajo la curva (representando la integral) y las líneas horizontales que indican el trabajo calculado por ambos métodos. La gráfica también muestra la diferencia porcentual en el título.

```
# Graficar la función de fuerza
plt.plot(x, fuerza(x), label='fuerza', color='b')
plt.fill_between(x, fuerza(x), alpha=0.3, label='Área bajo la curva (Trabajo Integral)')

# Agregar líneas horizontales para el trabajo calculado
plt.axhline(y=trabajo_integral, color='g', linestyle='--', label=f'Trabajo (Integral): {trabajo_integral:.4f}')
plt.axhline(y=trabajo_sumatoria, color='r', linestyle='--', label=f'Trabajo (Sumatoria): {trabajo_sumatoria:.4f}')
```

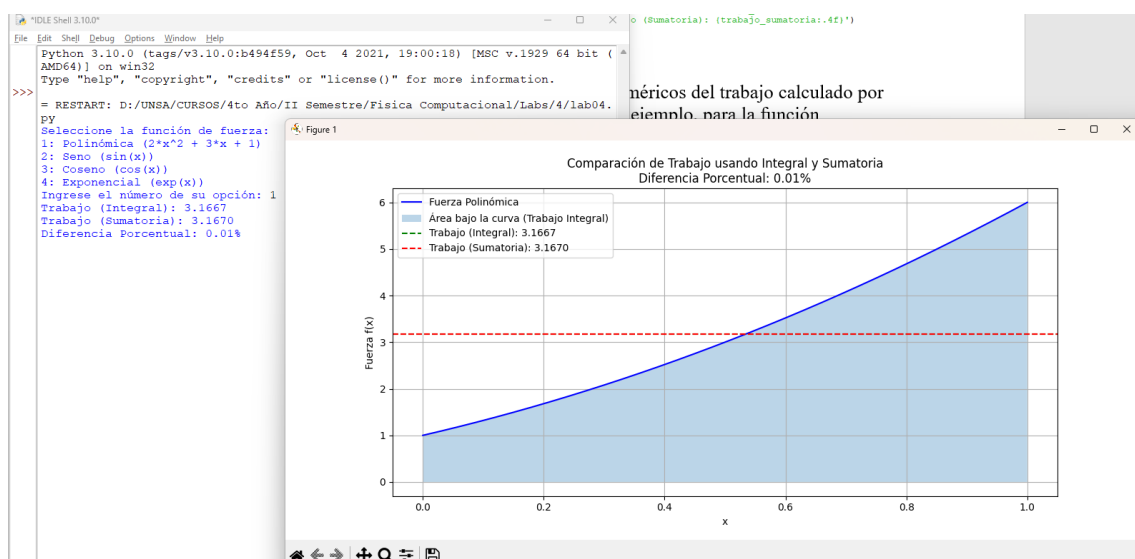
## Resultados

En cada ejecución, el código muestra los resultados numéricos del trabajo calculado por ambos métodos, así como la diferencia porcentual. Por ejemplo, para la función polinómica  $F(x)=2x^2+3x+1$ , se obtuvo:

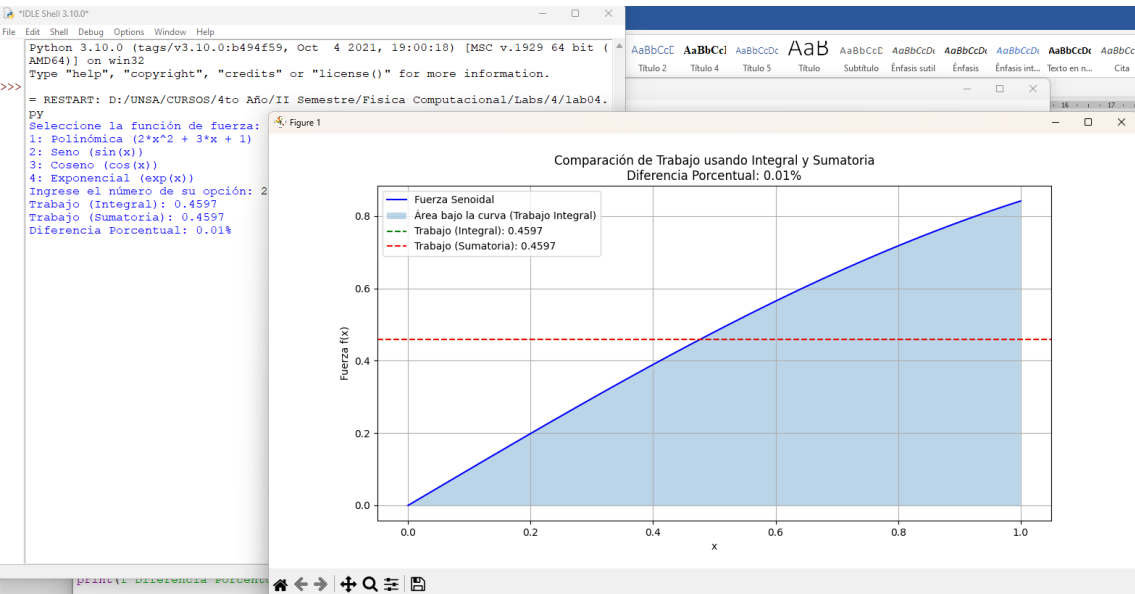
- Trabajo (Integral): 3.1667**
- Trabajo (Sumatoria): 3.1670**
- Diferencia porcentual: 0.01%**

La gráfica muestra la función de fuerza seleccionada y el área bajo la curva correspondiente a la integral. Las líneas horizontales indican los valores de trabajo obtenidos con ambos métodos, resaltando la pequeña diferencia entre ellos.

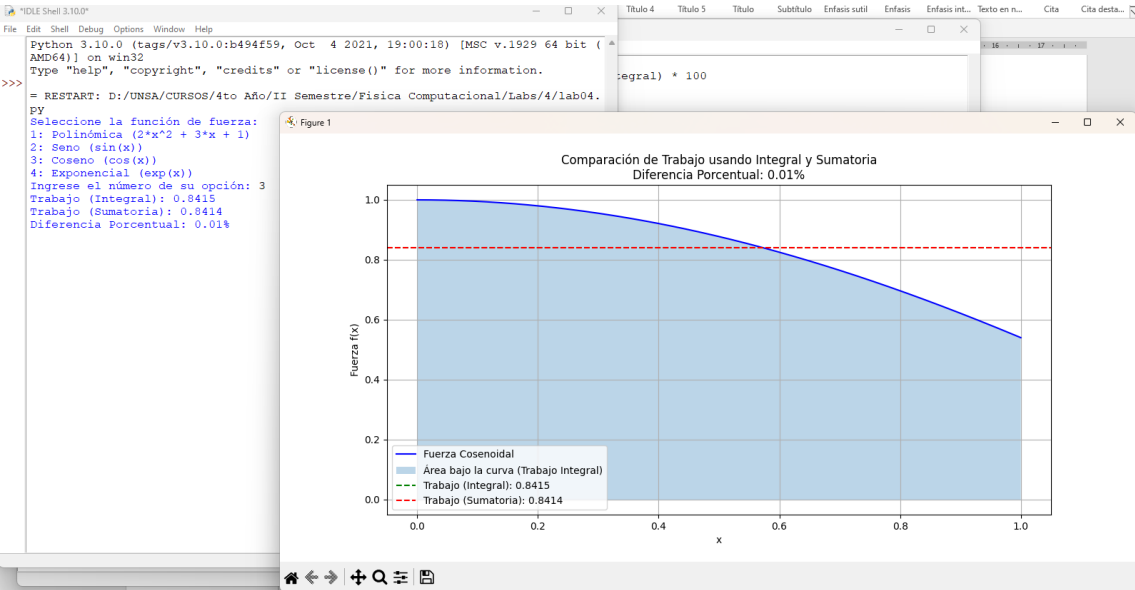
## FUNCION POLINOMICA



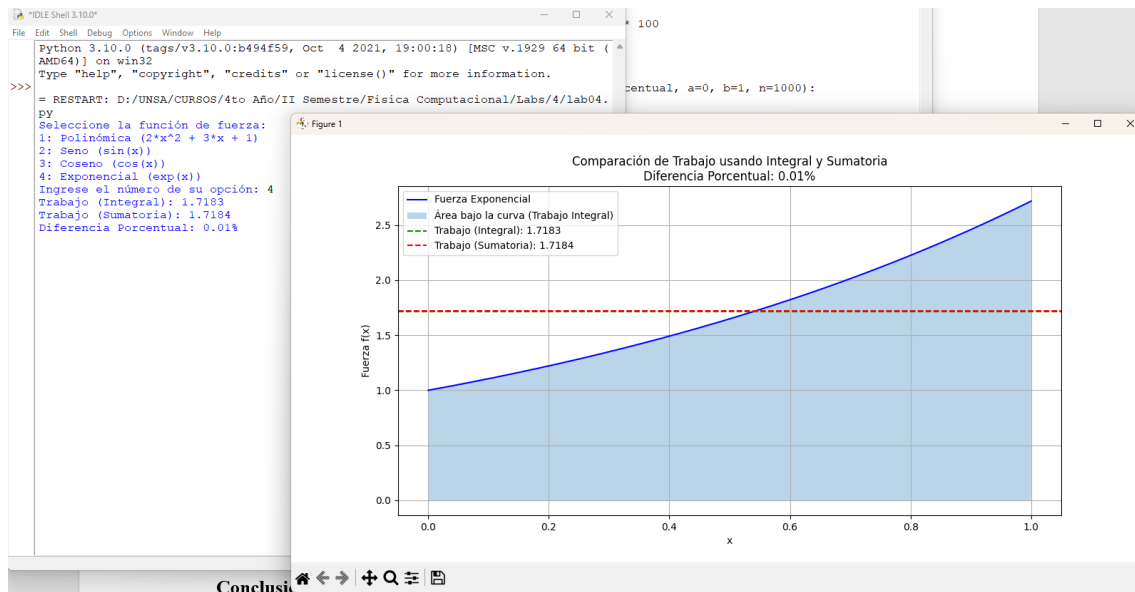
FUNCION SENO:



FUNCION COSENO:



FUNCION EXPONENCIAL



## Conclusiones

El cálculo del trabajo mediante la integral y la sumatoria muestra que la aproximación por sumatoria es bastante precisa, con una diferencia porcentual generalmente menor al 1%. Esto confirma la efectividad de la sumatoria para aproximar el trabajo, especialmente con un número alto de subdivisiones. La metodología aplicada es adaptable a diferentes funciones de fuerza, permitiendo un análisis versátil de problemas físicos relacionados con el trabajo mecánico.