INTEGRACIÓN COMPUTACIONAL

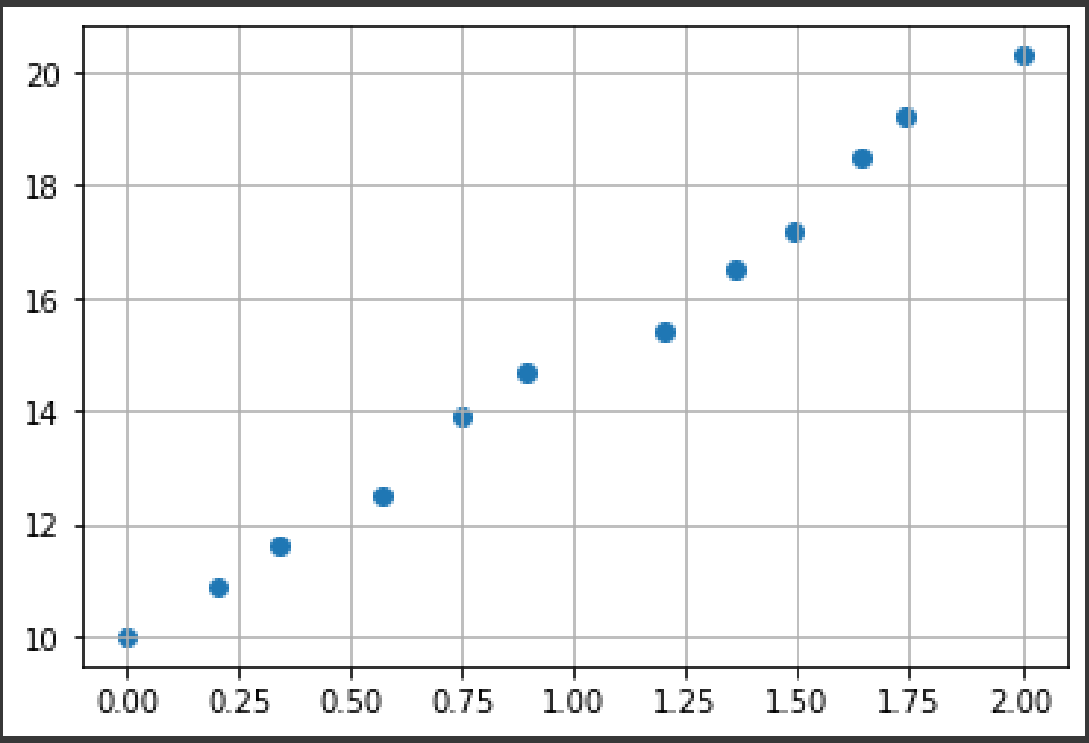
Universidad de Valparaíso CHILE

✓Ejercicio 1:

1. Dentro del archivo **datos.txt** se encuentran dos columnas de datos. La primera corresponde a mediciones del eje X, y la segunda corresponde a mediciones del eje Y. Utilizando dichos datos:

   a) Graficar los datos como puntos discretos
   b) Integrar los datos discretos por método Trapezoidal
   c) Integrar los datos discretos por método de Simpson.

[2]
```
import matplotlib.pyplot as pl
import numpy as np
from scipy import integrate

#PARTE A:

lista_x = np.loadtxt('datos.txt', float, usecols=0)
lista_y = np.loadtxt('datos.txt', float, usecols=1)

pl.scatter(lista_x, lista_y)
pl.grid()
pl.show()

#PARTE B:

resultado_trapezoidal = integrate.trapz(lista_y, lista_x)
print('Resultado por Trapezoidal es: ', resultado_trapezoidal)

#PARTE C:

resultado_simpson = integrate.simps(lista_y, lista_x)
print('Resultado por Simpson es :', resultado_simpson)
```




```
Resultado por Trapezoidal es:  29.92
Resultado por Simpson es : 29.917477945301545
```

Universidad de Valparaíso CHILE

✓Ejercicio 2:

Sea la función $f(x) = (2 + 3\sin(2x))^3$, integrada desde $x = 0$ a $x = \frac{\pi}{2}$

a) Calcular valor de la integral por método de Romberg
b) Calcular el valor de la integral por método de Cuadratura Gaussiana. Además, explicitar cuál es el error estimado del resultado encontrado.

```python
from scipy import integrate
import numpy as np

#PARTE A:

def f(x):
    return (2 + 3*np.sin(2 * x))**3

a = 0; b = np.pi / 2

resultado_romberg = integrate.romberg(f, a, b, show=False)
print('Resultado por Romberg: ', resultado_romberg)

#PARTE B:

output_gauss = integrate.quadrature(f, a, b)
resultado_gauss = output_gauss[0]
error_gauss = output_gauss[1]

print(' ')
print('El resultado por Cuadratura Gaussiana es: ', resultado_gauss)
print('Su error estimado es: ', error_gauss)
```

```
Resultado por Romberg:  108.97787143714103

El resultado por Cuadratura Gaussiana es:  108.97787144745811
Su error estimado es:  5.249702752507801e-07
```

INTERPOLACIÓN

Universidad de Valparaíso CHILE

✓Ejercicio 3:

Dentro del archivo mediciones.txt se encuentran dos columnas de datos. La primera corresponde a mediciones del eje X, y la segunda corresponde a mediciones del eje Y. Utilizando dichos datos:

a) Grafique los datos como puntos discretos
b) Interpolarlos para obtener una función polinómica p(x) que atraviese todos los puntos
c) Graficar la función polinómica p(x) sobre los datos discretos

```python
import numpy as np
import matplotlib.pyplot as pl

#PARTE A:

lista_x = np.loadtxt('mediciones.txt', float, usecols = 0)
lista_y = np.loadtxt('mediciones.txt', float, usecols = 1)

pl.scatter(lista_x, lista_y, color = 'red')
#También se podría poner el scatterplot junto con los demás.

#PARTE B:
cantidad_de_datos = len(lista_x)

lista_de_coeficientes = np.polyfit(lista_x, lista_y, cantidad_de_datos - 1)

p = np.poly1d(lista_de_coeficientes) #Ya tenemos una función p(x)

#PARTE C:

#Hacer que hayan más valores entre el primer y último valor en la lista x para generar una curva más suave.
primer_x = lista_x[0]
ultimo_x = lista_x[-1]

lista_x_extendida = np.linspace(primer_x, ultimo_x, 100)
pl.plot(lista_x_extendida, p(lista_x_extendida))
#pl.legend((p,), loc='best')
pl.grid()
pl.show()

#La gracia de esto es que ahora podemos saber valores que antes no conocíamos.
#Por ejemplo
#print(p(3.5))
```