Universidad de Valparaíso
CHILE

AYUDANTÍA FÍSICA COMPUTACIONAL II

Ayudante: Nicolás Gatica

Correo: nicolas.gatica@alumnos.uv.cl

INTEGRACIÓN COMPUTACIONAL

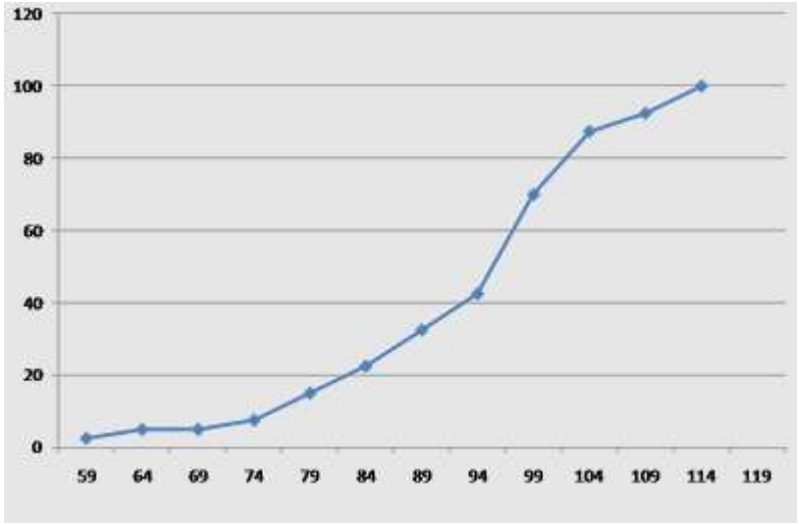
Universidad de Valparaíso CHILE

- ✓Errores y límites computacionales
- ✓Integración por método de Trapezoidal
- ✓Integración por método de Simpson
- ✓Integración por método de Romberg y cuadratura Gaussiana

INTEGRACIÓN COMPUTACIONAL

Universidad de Valparaíso CHILE

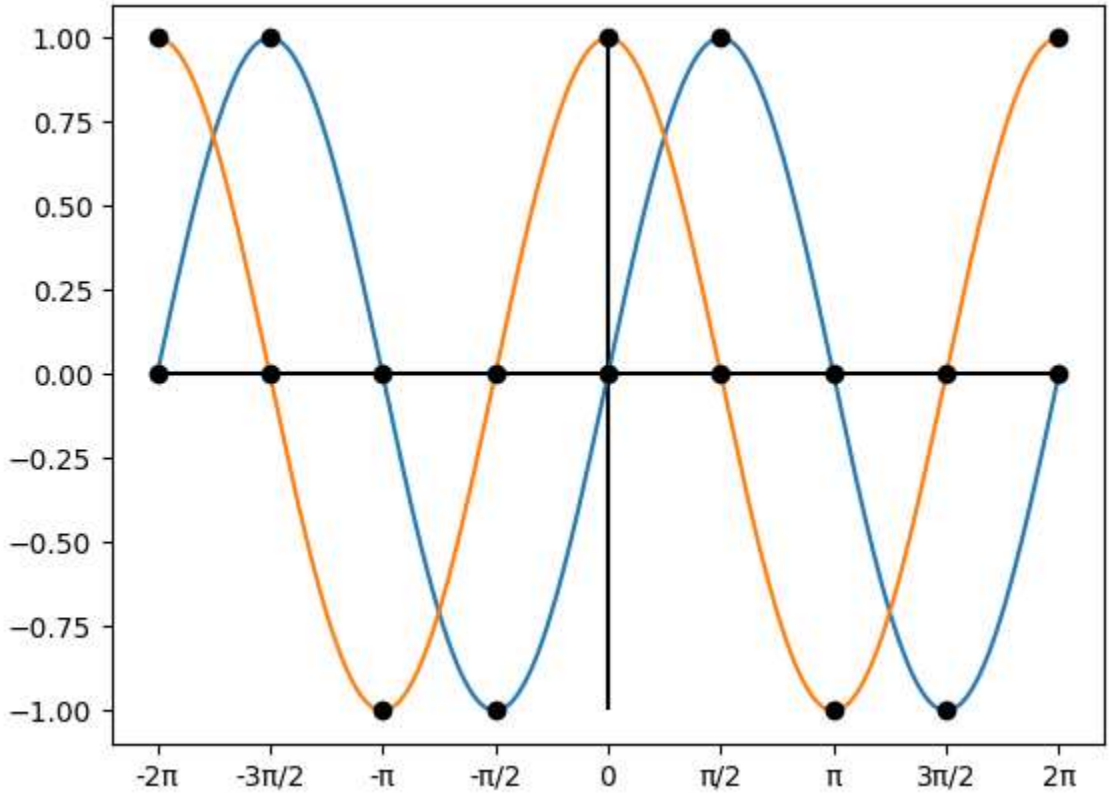
El método por Trapezoidal y Simpson es útil al querer integrar datos discretos en el que no conocemos la función.



INTEGRACIÓN COMPUTACIONAL

Universidad de Valparaíso CHILE

Por otra parte, el método de Romberg y cuadratura Gaussiana son útiles para integrar funciones definidas.



INTEGRACIÓN COMPUTACIONAL

Universidad de Valparaíso CHILE

La dificultad reside en saber llevar las expresiones matemáticas al código.

Trapezoidal: $I = h\left[\frac{f(a)+f(b)}{2} + \sum_{i=1}^{n-1} f(a + h * i)\right]$

Simpson: $I = \frac{h}{3}\left[f(a) + 4\sum_{1,3,5}^{n-1} f(a + hi) + 2\sum_{2,4,6}^{n-2} f(a + hi) + f(b)\right]$

INTEGRACIÓN COMPUTACIONAL

Universidad de Valparaíso CHILE

Aunque puede resultar útil saber programar este tipo de integrales a mano, ya hay funciones en Python que hacen esto mismo y mejor.

Dentro del paquete scipy se encuentra la librería integrate:

```
>>>from scipy import integrate #import scipy.integrate as sc
>>>lista_y = [1, 1.5, 3, … , n]
>>>lista_x = [-4, 52, 0.075, … , n]
>>>I_trapz = integrate.trapz(lista_y, lista_x) #Entrega resultado numérico
>>>I_Simp = integrate.simps(lista_y, lista_x) #Entrega resultado numérico
```

INTEGRACIÓN COMPUTACIONAL

Universidad de Valparaíso CHILE

✓Ejercicio 1:

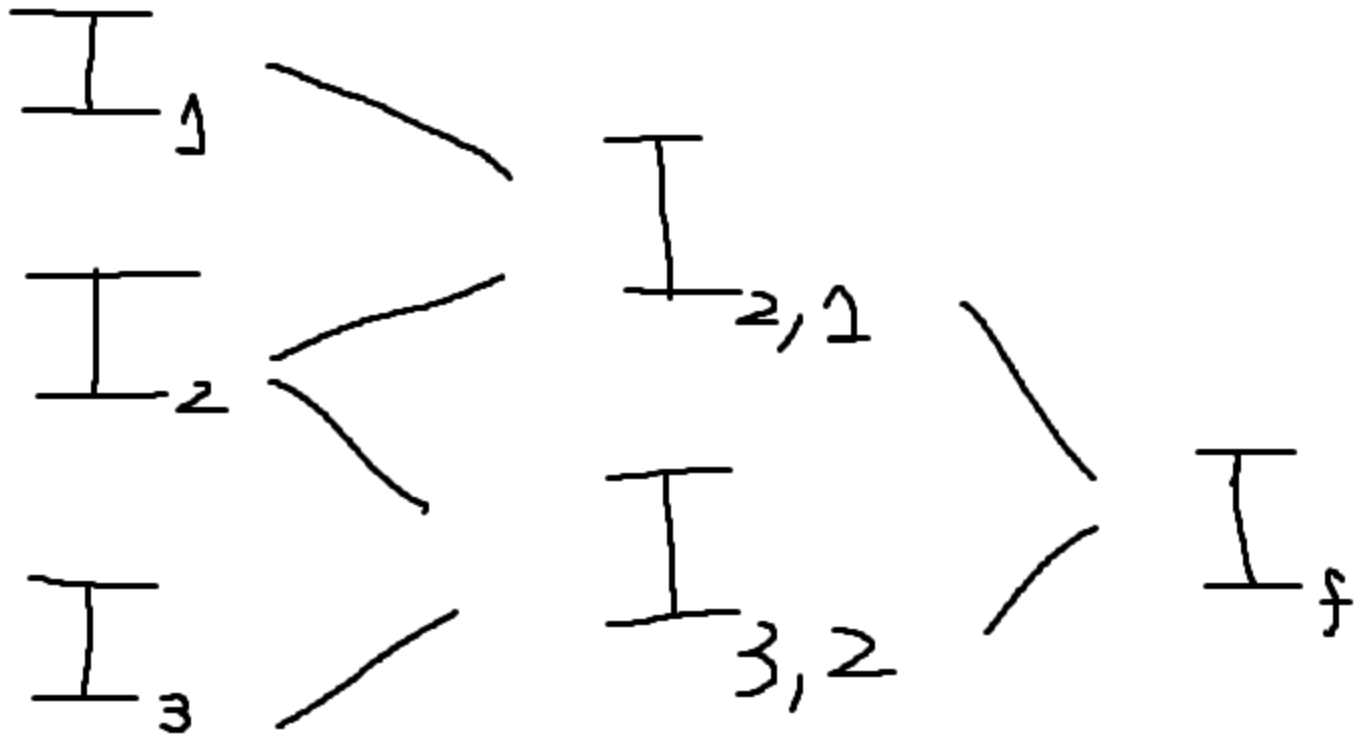
1. Dentro del archivo **datos.txt** se encuentran dos columnas de datos. La primera corresponde a mediciones del eje X, y la segunda corresponde a mediciones del eje Y. Utilizando dichos datos:

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

INTEGRACIÓN COMPUTACIONAL

Universidad de Valparaíso CHILE

Método de Romberg permite que a partir de los resultados de integrales, pueden obtener más resultados sin hacer la integral.



INTEGRACIÓN COMPUTACIONAL

Universidad de Valparaíso CHILE

Al igual que con Trapezoidal y Simpson, ya hay funciones en python que calculan esto sin necesidad de hacerlo a mano.

Dentro del paquete scipy se enceuntra la librería integrate:

```
>>>integrate.Romberg(f, a, b, show=True/False)
```

f: función

a,b: Límites de integración

show: Muestra el triangulo de Romberg

INTEGRACIÓN COMPUTACIONAL

Universidad de Valparaíso CHILE

Por ultimo, el método de Cuadratura Gaussiana es más exacto que Romberg. La teoría matemática detrás de este método es algo engorroso.

>>>integrate.quadrature(f, a, b)

f: función

a,b: Límites de integración

→Entrega un array con dos elementos.

>>>print(integrate.quadrature(f, a, b))

>>>[resultado, error]

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.

INTEGRACIÓN COMPUTACIONAL

Universidad de Valparaíso CHILE

En física, es común que los problemas sean multivariable. Esto quiere decir al integrar funciones multivariable, tendrémos integrales dobles, triples o hasta cuadruples.

Afortunadamente, también hay funciones ya hechas en python para estos casos.

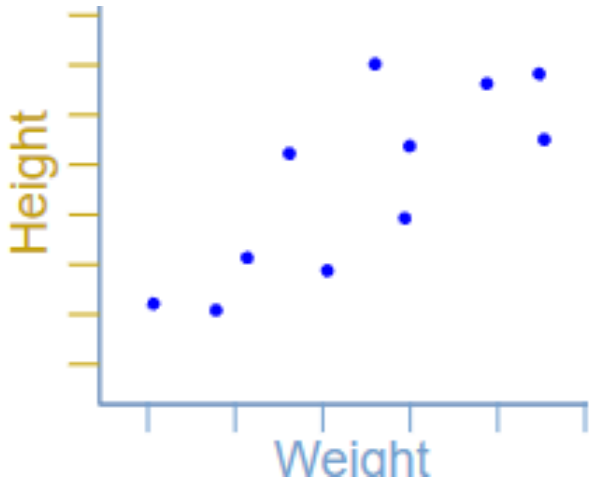
$$\int_0^2 \int_0^1 xy^2 \, dydx$$

```
>>>import scipy.integrate as sc
>>>def f(y,x):
        return x * y**2
>>>I_2d = sc.nquad(f, [[0, 1], [0, 2]])
```

Entrega un array con el valor de la integral y el error.

INTERPOLACIÓN

La interpolación corresponde a encontrar una función f(x) que se ajuste a todos los puntos.



INTERPOLACIÓN

Universidad de Valparaíso CHILE

El proceso de encontrar una función que no conocemos se llama ecuación diferencial ordinaria, parcial o estocástica. Sin embargo, solemos trabajar mayormente con las dos primeras.

Ejemplos:

$$\frac{dy}{dx} = x^3 - 6$$

$$y'' + y' + 5y = e^x$$

$$y' - 2\frac{y}{x} = 0$$

INTERPOLACIÓN

Universidad de Valparaíso CHILE

Gracias a la computación, podemos encontrar funciones que se ajusten a las curvas sin necesidad de recurrir a métodos de ecuaciones diferenciales o parciales.

Dentro del paquete numpy se encuentra polyfit:

```
>>>from numpy import polyfit, poly1d
>>>lista_y = [1, 1.5, 3, … , n]
>>>lista_x = [-4, 52, 0.075, … , n]
>>>a = polyfit(lista_x, lista_y, grado)
>>>#Polyfit entrega un array de coeficientes
```

INTERPOLACIÓN

Universidad de Valparaíso CHILE

Dado a que la función anterior sólo nos entrega coeficientes, pareciera que quedamos donde mismo. Sin embargo, hay otra función que complementa a la anterior dentro de numpy, la cuál es poly1d.

```
>>>from numpy import polyfit, poly1d
>>>lista_x = [-4, 52, 0.075, … , n]
>>>lista_y = [1, 1.5, 3, … , n]
>>>grado = len(lista_x)
>>>Coef = polyfit(lista_x, lista_y, grado - 1)
>>>p = poly1d(Coef)
>>>#poly1d define una función polinómica cuyos coeficientes.
```

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

AJUSTE DE CURVAS

Universidad de Valparaíso CHILE

La interpolación puede resultar muy útil, sin embargo, asumimos que nuestros datos son exactos, los cuales en general no lo son. Sin embargo, esto puede ser rápidamente solucionado al ajustar una curva que más o menos describa como se comportan estos datos discretos.

```
>>>from numpy import polyfit, poly1d
>>>lista_x = [-4, 52, 0.075, … , n]
>>>lista_y = [1, 1.5, 3, … , n]
>>>Coef = polyfit(lista_x, lista_y, grado_que_uno_quiera)
>>>p = poly1d(Coef)
```

AJUSTE DE CURVAS

Universidad de Valparaíso CHILE

✓Ejercicio 4:

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) Ajustar a los datos una curva polinómica de **primer grado** y graficarla sobre los puntos discretos
c) Ajustar a los datos una curva polinómica de **segundo grado** y graficarla sobre los puntos discretos
d) Ajustar a los datos una curva polinómica de **tercer grado** y graficarla sobre los puntos discretos