

Ayudantía Física Computacional II: Resumen para 2da Prueba

Integración de datos discretos

from **scipy** import **integrate**

- a) **Por método Trapezoidal:**
`integrate.trapz(lista_y, lista_x)`

¿Qué entrega?:
Resultado numérico de la integral.

- b) **Por método Simpson:**
`integrate.simps(lista_y, lista_x)`

¿Qué entrega?:
Resultado numérico de la integral.

Integración de funciones definidas

from **scipy** import **integrate**

- a) **Por método de Romberg:**
`integrate.romberg(f, a, b, show = True/False)`

¿Qué entrega?:
Depende. Si se colocó **True**, entrega la pirámide de Romberg completa. Si se colocó **False**, entrega solamente el mejor resultado numérico de la integral.

- b) **Por método de Cuadratura Gaussiana:**
`integrate.quadrature(f, a, b)`

¿Qué entrega?:
Un array con dos elementos: el primero corresponde al resultado numérico de la integral, y el segundo corresponde al tamaño del error estimado de aquel resultado numérico.

f: función matemática definida previamente.
a: límite inferior de integración.
b: límite superior de integración.

Interpolación

from **numpy** import **polyfit, poly1d**

1. **Encontrando coeficientes:**
`polyfit(lista_x, lista_y, grado)`

¡OJO!: **grado** debe ser igual a la **cantidad** de puntos discretos, **restándole 1**.

¿Qué entrega?:
Array con los coeficientes de una función polinómica que atraviesa todos los puntos discretos.

2. **Definiendo la función polinómica:**
`poly1d(array_de_coeficientes)`

¿Qué entrega?:
Define una función polinómica, cuyos coeficientes son los ingresados.

Ajuste de Curvas

from **numpy** import **polyfit, poly1d**

1. **Encontrando coeficientes:**
`polyfit(lista_x, lista_y, grado)`

¡OJO!: **grado** puede ser igual a cualquier número. Depende de si queremos ajustar una curva cuyo grado sea 1, 2, 3, o el que sea.

¿Qué entrega?:
Array con los coeficientes de una función polinómica que no necesariamente atraviesa a todos los puntos discretos, pero que más o menos los representa

2. **Definiendo la función polinómica:**
`poly1d(array_de_coeficientes)`

¿Qué entrega?:
Define una función polinómica, cuyos coeficientes son los ingresados.

Spline Lineal

Hay dos alternativas para efectuar un Spline Lineal:

- a) **Usando la función de numpy:**
from **numpy** import **interp**

`interp(lista_x_poblado, lista_x, lista_y)`

¿Qué entrega?:
Lista de valores del eje Y para ser graficados junto con **lista_x_poblado**, y así dibujar el Spline Lineal.

- b) **Usando la función de scipy:**
from **scipy** import **interpolate**

`interpolate.interp1d(lista_x, lista_y)`

¿Qué entrega?:
Define una función matemática, la cual puede ser utilizada para graficar el Spline Lineal.

Spline Cúbico

from **scipy** import **interpolate**

`interpolate.interp1d(lista_x, lista_y, kind = 'cubic')`

¿Qué entrega?:
Define una función matemática, la cual puede ser utilizada para graficar el Spline Cúbico.