

## Computación

Aula Virtual: <https://famaf.aulavirtual.unc.edu.ar/course/view.php?id=747>

Resguardo tutoriales: <https://www.famaf.unc.edu.ar/~moreschi/docencia/Computacion/>

### Tutorial Problema 9 de la Guía N° 4

#### Problema 9:

Interpolación de Lagrange. Escriba un script de PYTHON que implemente la interpolación polinómica en forma de Lagrange. Dadas dos listas de números float de igual longitud, que representan los valores  $x_i$  e  $y_i$ ,  $i = 0, 1, \dots, N$  a interpolar, el script debe definir una función `ilagrangle` que calcula el valor del polinomio interpolatorio en el punto  $x$ . Pruebe su script realizando la interpolación de la función  $\sin(x)$ , evaluando en los puntos  $[0, \frac{\pi}{4}, \frac{2\pi}{4}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{6\pi}{4}, \frac{7\pi}{4}, 2\pi]$ . Calcule el valor del polinomio interpolatorio en el punto  $x = \frac{\pi}{6}$ .

---

#### Tutorial:

- Guarde en el archivo `p9.py` las siguientes instrucciones:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def ilagrangle(x, xp, fp):
5     pol = 0.0
6     # suma sobre los puntos
7     for m in range(len(xp)):
8         l = 1.
9         for n in range(len(xp)):
10            if n == m:
11                continue
12            l *= (x - xp[n]) / (xp[m] - xp[n])
13        pol += fp[m] * l
14    return pol
15
16 xmin = 0.0
17 xmax = 2.0 * np.pi
18
19 N = 9
20 xp = np.linspace(xmin, xmax, N)
21 fp = np.sin(xp)
22
23 xx = np.linspace(xmin, xmax, 200)
24 ff = np.zeros(len(xx))
25
26 for n in range(len(xx)):
27     ff[n] = ilagrangle(xx[n], xp, fp)
28
29 fexact = np.sin(xx)
30
31 # error
```

```

32 e = fexact - ff
33
34 plt.figure( figsize=(12, 8))
35 plt.subplot(211)
36 plt.title('sin(x) y polinomio de Lagrange')
37 plt.scatter(xp, fp, marker="x", color="r", s=30, label='puntos')
38 plt.plot(xx, fexact, color="cyan", linewidth=3.5, label='exacta')
39 plt.plot(xx, ff, color="k", label='polinomio')
40 plt.legend(loc='best')
41 plt.xlabel('x')
42 plt.ylabel('y')
43 plt.grid()
44 plt.xlim(xmin, xmax)
45 plt.axhline(y=0, color='#000000') # línea horizontal
46
47 plt.subplot(212)
48 plt.title('Error')
49 plt.plot(xx, e, label='error')
50 plt.legend(loc='best')
51 plt.xlabel('x')
52 plt.ylabel('y')
53 plt.grid()
54 plt.xlim(xmin, xmax)
55 plt.axhline(y=0, color='#000000') # línea horizontal
56
57 plt.subplots_adjust(hspace=0.3)
58 plt.savefig("graficos/p9-lagrange.png")
59 plt.show()
60
61 print()
62 print('ilagrange(np.pi/6., xp, fp) =', ilagrange(np.pi/6., xp, fp))
63 print('      np.sin(np.pi/6.) =', np.sin(np.pi/6.))
64 print()

```

- Desde la terminal ejecute:

```
python3 p9.py
```

e interprete el resultado.

Alternativamente ejecute:

```
python3
```

y vaya agregando uno a uno los bloques del programa.

Obtendrá algo como esto:

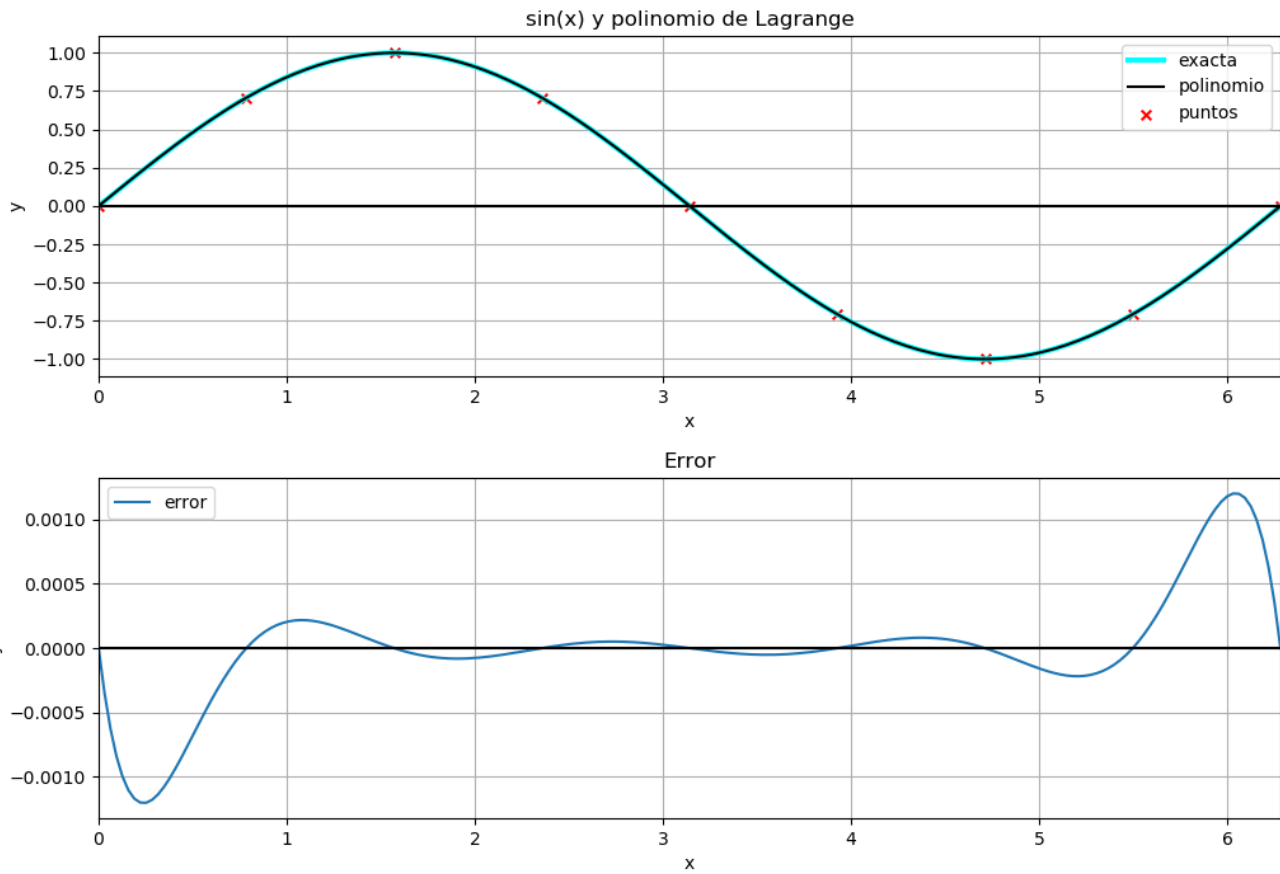


Figura 1: Funciones y error.

```

1 ilagrange(np.pi/6., xp, fp) = 0.5006139353456853
2 np.sin(np.pi/6.) = 0.49999999999999994

```