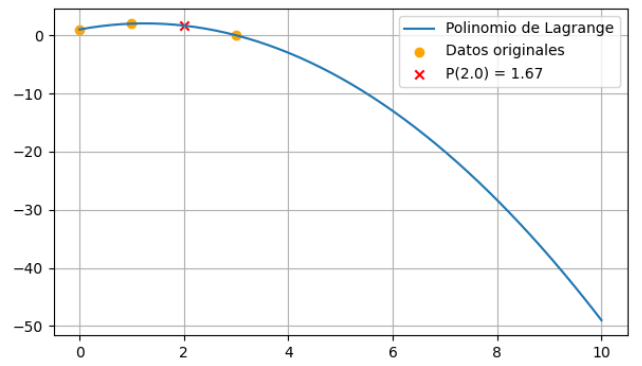


Actividad 1 – II Unidad: Introducción a la aproximación computacional (Interpolación Lagrange)

EJERCICIO 2

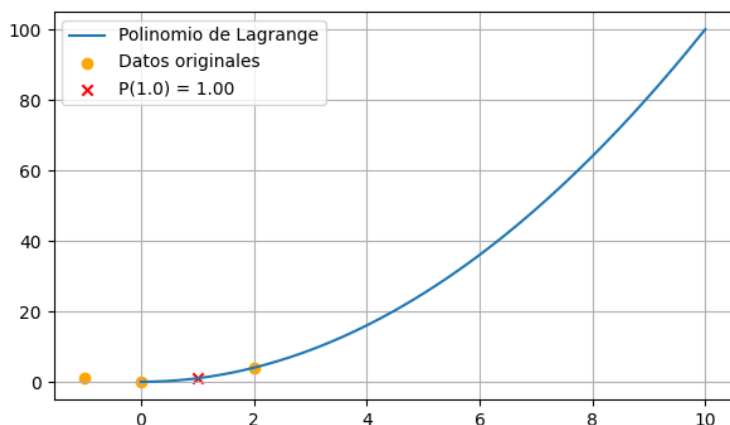
```
1 # =====
2 # EJERCICIO 2 - Interpolación de Lagrange
3 # Puntos: (0,1), (1,2), (3,0)
4 # =====
5
6 import numpy as np
7 import matplotlib.pyplot as plt
8
9 # Datos
10 x_points = np.array([0.0, 1.0, 3.0])
11 y_points = np.array([1.0, 2.0, 0.0])
12 x_eval = 2.0
13
14 # ---- Función para obtener coeficientes del polinomio de Lagrange ----
15 def lagrange_poly_coeffs(xs, ys):
16     n = len(xs)
17     result = np.zeros(n)
18     for i in range(n):
19         xs_excl = np.delete(xs, i)
20         numer_coeffs = np.poly(xs_excl)
21         denom = np.prod(xs[i] - xs_excl)
22         if len(numer_coeffs) < len(result):
23             numer_coeffs = np.pad(numer_coeffs, (len(result) - len(numer_coeffs), 0), 'constant')
24         result = result + ys[i] * numer_coeffs / denom
25     return result
26
27 # ---- Construir polinomio y evaluar ----
28 coeffs = lagrange_poly_coeffs(x_points, y_points)
29 P = np.poly1d(coeffs)
30 y_eval = P(x_eval)
31 print("=== EJERCICIO 2 ===")
32 print("Polinomio interpolador P(x):")
33 print(P)
34 print(f"P({x_eval}) = {y_eval:.6f}")
35
36 # ---- Graficar ----
37 x_plot = np.linspace(0, 10, 200)
38 y_plot = P(x_plot)
39
40 plt.figure(figsize=(7,4))
41 plt.plot(x_plot, y_plot, label='Polinomio de Lagrange')
42 plt.scatter(x_points, y_points, color='orange', label='Datos originales')
43 plt.scatter([x_eval], [y_eval], color='red', marker='x', label=f"P({x_eval}) = {y_eval:.2f}")
44 plt.grid(True)
45 plt.legend()
46 plt.show()
```



Grafica Interpretación: Se puede observar en la gráfica como el polinomio primero crece y luego disminuye esto debido a que el polinomio cuadrático no presenta una relación lineal respecto a X y Y.

EJERCICIO 3

```
1 # =====
2 # EJERCICIO 3 - Interpolación de Lagrange
3 # Puntos: (-1,1), (0,0), (2,4)
4 # =====
5 import numpy as np
6 import matplotlib.pyplot as plt
7
8 # Datos
9 x_points = np.array([-1.0, 0.0, 2.0])
10 y_points = np.array([1.0, 0.0, 4.0])
11 x_eval = 1.0 |
12
13 # ---- Función para obtener coeficientes del polinomio de Lagrange ----
14 def lagrange_poly_coeffs(xs, ys):
15     n = len(xs)
16     result = np.zeros(n)
17     for i in range(n):
18         xs_excl = np.delete(xs, i)
19         numer_coeffs = np.poly(xs_excl)
20         denom = np.prod(xs[i] - xs_excl)
21         if len(numer_coeffs) < len(result):
22             numer_coeffs = np.pad(numer_coeffs, (len(result) - len(numer_coeffs), 0), 'constant')
23         result = result + ys[i] * numer_coeffs / denom
24     return result
25
26 # ---- Construir polinomio y evaluar ----
27 coeffs = lagrange_poly_coeffs(x_points, y_points)
28 P = np.poly1d(coeffs)
29 y_eval = P(x_eval)
30
31 print("=== EJERCICIO 3 ===")
32 print("Polinomio interpolador P(x):")
33 print(P)
34 print(f"P({x_eval}) = {y_eval:.6f}")
35 # ---- Graficar ----
36 x_plot = np.linspace(0, 10, 200)
37 y_plot = P(x_plot)
38
39 plt.figure(figsize=(7,4))
40 plt.plot(x_plot, y_plot, label='Polinomio de Lagrange')
41 plt.scatter(x_points, y_points, color='orange', label='Datos originales')
42 plt.scatter([x_eval], [y_eval], color='red', marker='x', label=f"P({x_eval}) = {y_eval:.2f}")
43 plt.grid(True)
44 plt.legend()
45 plt.show()
46
```



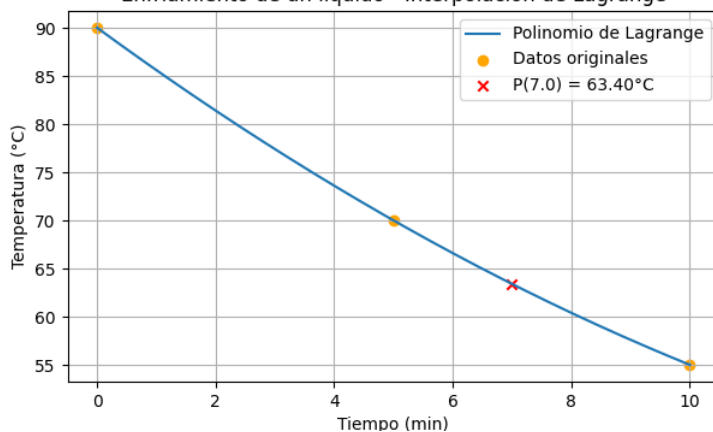
Interpretación: La gráfica muestra un comportamiento **parabólico ascendente**, característico de procesos donde la variable dependiente crece aceleradamente con el tiempo o el valor de entrada, como una función de energía o distancia cuadrática.

EJERCICIO 5:

- Programa en python

```
1 # EJERCICIO 5 - Interpolación de Lagrange
2 # Enfriamiento de un líquido
3 # Puntos: (0,90), (5,70), (10,55)
4 import numpy as np
5 import matplotlib.pyplot as plt
6 # ---- Datos ----
7 x_points = np.array([0.0, 5.0, 10.0]) # Tiempo (min)
8 y_points = np.array([90.0, 70.0, 55.0]) # Temperatura (°C)
9 x_eval = 7.0 # Evaluar a los 7 minutos
10
11 # ---- Función para obtener coeficientes del polinomio de Lagrange ----
12 def lagrange_poly_coeffs(xs, ys):
13     n = len(xs)
14     result = np.zeros(n)
15     for i in range(n):
16         xs_excl = np.delete(xs, i)
17         numer_coeffs = np.poly(xs_excl)
18         denom = np.prod(xs[i] - xs_excl)
19         if len(numer_coeffs) < len(result):
20             numer_coeffs = np.pad(numer_coeffs, (len(result) - len(numer_coeffs), 0), 'constant')
21         result = result + ys[i] * numer_coeffs / denom
22     return result
23 # ---- Construir polinomio y evaluar ----
24 coeffs = lagrange_poly_coeffs(x_points, y_points)
25 P = np.poly1d(coeffs)
26 y_eval = P(x_eval)
27
28 print("=== EJERCICIO 5 ===")
29 print("Polinomio interpolador P(x):")
30 print(P)
31 print(f"P({x_eval}) = {y_eval:.6f} °C")
32
33 # ---- Graficar ----
34 x_plot = np.linspace(0, 10, 200)
35 y_plot = P(x_plot)
36
37 plt.figure(figsize=(7,4))
38 plt.plot(x_plot, y_plot, label='Polinomio de Lagrange')
39 plt.scatter(x_points, y_points, color='orange', label='Datos originales')
40 plt.scatter([x_eval], [y_eval], color='red', marker='x', label=f"P({x_eval}) = {y_eval:.2f}°C")
41 plt.title("Enfriamiento de un líquido - Interpolación de Lagrange")
42 plt.xlabel("Tiempo (min)")
43 plt.ylabel("Temperatura (°C)")
44 plt.grid(True)
45 plt.legend()
46 plt.show()
```

- Grafica
- Enfriamiento de un líquido - Interpolación de Lagrange



- Interpretación: La gráfica modela el enfriamiento de un líquido (por ejemplo, agua caliente) al paso del tiempo.

Al principio, la temperatura baja rápidamente (de 90 °C a 70 °C en los primeros 5 min).

Luego, la curva se aplana, mostrando que la pérdida de calor es más lenta conforme el líquido se acerca a la temperatura ambiente.

- Cálculo matemático

$$X_0 = 0 \quad Y_0 = 90 \quad X_1 = 5 \quad Y_1 = 70 \quad X_2 = 10 \quad Y_2 = 55$$

$$L_0(x) = \frac{(x-5)(x-10)}{(0-5)(0-10)} = L_0(x) = \frac{x^2 - 15x + 50}{50}$$

$$L_1(x) = \frac{(x-0)(x-10)}{(5-0)(5-10)} = L_1(x) = \frac{x^2 - 10x}{-25} = \frac{x^2 - 10x}{25}$$

$$L_2(x) = \frac{(x-0)(x-5)}{(10-0)(10-5)} = \frac{x^2 - 5x}{50}$$

$$P(x) = 90L_0 + 70L_1 + 55L_2$$

$$P(x) = \frac{1}{10}x^2 - \frac{9}{2}x + 90$$

$$P(7) = \frac{1}{10} \cdot 49 - \frac{9}{2} \cdot 7 + 90 = 4.9 - 31.5 + 90 = 63.4^\circ\text{C}$$