Tarea6_LeninAmangandi

November 26, 2025

Tarea Nº6

Métodos Númericos

Tema: Serie de Taylor y Polinomios de Lagrange

Nombre: Lenin Amangandi

Enlace GitHub Tarea 6

0.1 Ejercicio 1: Serie de Taylor

Función:

$$f(x) = \frac{1}{25x^2 + 1}$$

Punto de expansión:

$$x_0 = 0$$

- **Orden 0**:

$$P_0(x) = f(0) = \frac{1}{25(0)^2 + 1} = 1$$

- **Orden 1**: Derivada primera:

$$f'(x) = -\frac{50x}{(25x^2 + 1)^2}$$

Evaluada en

$$x_0 = 0$$

:

$$f'(0) = 0$$

Polinomio de orden 1:

$$P_1(x) = 1 + 0(x - 0) = 1$$

- **Orden 2**: Derivada segunda:

$$f''(x) = \frac{-50(25x^2 + 1)^2 + 50x(-100x)}{(25x^2 + 1)^4}$$

Evaluación en

$$x_0 = 0$$

:

$$f''(0) = -50$$

Polinomio de orden 2:

$$P_2(x) = 1 + 0(x-0) - 25(x-0)^2 = 1 - 25x^2$$

- **Orden 3**: Derivada tercera:

$$f'''(x) = \frac{6x^2 - 2}{(25x^2+1)^3}$$

Evaluación en

$$x_0 = 0$$

:

$$f'''(0) = \frac{6(0)^2 - 2}{(25(0)^2+1)^3} = \frac{-2}{1} = -2$$

Polinomio de orden 3:

$$P_3(x) = 1 - 25x^2 + (-2)\,(x-0)^3 = 1 - 25x^2 - 2x^3$$

- **Orden 4**: Derivada cuarta:

$$f^{(4)}(x) = \frac{-24x(x^2-1)}{(25x^2+1)^4}$$

Evaluación en

$$x_0 = 0$$

:

$$f^{(4)}(0) = 0$$

Polinomio de orden 4:

$$P_4(x) = 1 - 25x^2 - 2x^3 + 0 \cdot x^4 = 1 - 25x^2 - 2x^3$$

- **Orden 5**: Derivada quinta:

$$f^{(5)}(x) = \frac{-24(x^4 - 2x^2 + 1)}{(25x^2+1)^5}$$

Evaluación en

$$x_0 = 0$$

:

$$f^{(5)}(0) = 0$$

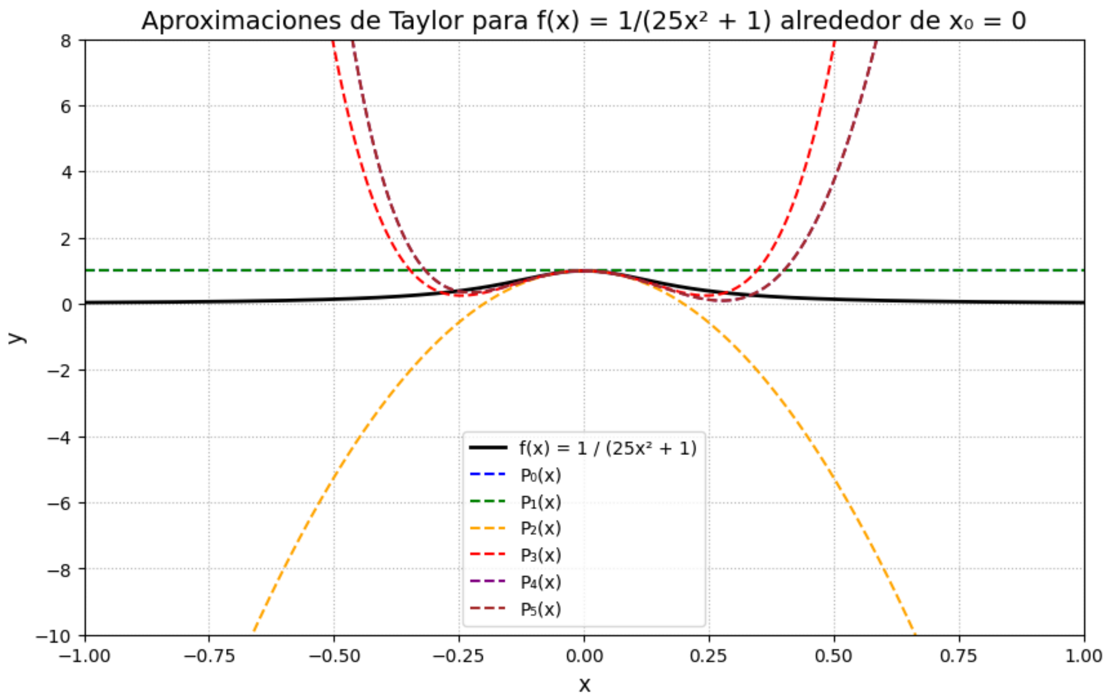
Polinomio de orden 5:

$$P_5(x) = 1 - 25x^2 - 2x^3 + 0 \cdot x^4 + 0 \cdot x^5 = 1 - 25x^2 - 2x^3$$

[9]:

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

x_min = -1
x_max = 1
num_puntos = 600
x0 = 0

def f1(x):
    return 1 / (25 * x**2 + 1)

def P0(x):
    return np.ones_like(x) * 1

def P1(x):
    return np.ones_like(x) * 1

def P2(x):
    return 1 - 25 * x**2

def P3(x):
    return 1 - 25 * x**2 + (625/3) * x**4

def P4(x):
    return 1 - 25 * x**2 + (625/3) * x**4 - (3125/24) * x**5

def P5(x):
    return 1 - 25 * x**2 + (625/3) * x**4 - (3125/24) * x**5

x = np.linspace(x_min, x_max, num_puntos)

plt.figure(figsize=(10, 6))

plt.plot(x, f1(x), 'k', label='f(x) = 1 / (25x² + 1)', linewidth=2)

plt.plot(x, P0(x), '--', color='blue', label='P (x)')
plt.plot(x, P1(x), '--', color='green', label='P (x)')
plt.plot(x, P2(x), '--', color='orange', label='P (x)')
plt.plot(x, P3(x), '--', color='red', label='P (x)')
plt.plot(x, P4(x), '--', color='purple', label='P (x)')
plt.plot(x, P5(x), '--', color='brown', label='P (x)')

plt.title("Aproximaciones de Taylor para f(x) = 1/(25x² + 1) alrededor de x  = 0", fontsize=14)
plt.xlabel("x", fontsize=12)
plt.ylabel("y", fontsize=12)
plt.grid(True, linestyle=':')
```

```
plt.legend()

plt.ylim(-10, 8)

plt.xlim(x_min, x_max)
plt.show()
```



0.2 Ejercicio 1: Serie de Lagrange

Ejercicio: Aproximación de la función

$$f(x) = \frac{1}{25x^2 + 1}$$

mediante la Interpolación de Lagrange

Consideramos los puntos

$$x_0 = -0.5, x_1 = 0, x_2 = 0.5$$

como nuestros puntos de interpolación.

1. Calcular los valores de la función en los puntos (x_0, x_1, x_2):

-
$$f(x_0) = f(-0.5) = \frac{1}{25(-0.5)^2 + 1} = \frac{1}{25(0.25) + 1} = \frac{1}{7.25}$$

-
$$f(x_1) = f(0) = \frac{1}{25(0)^2 + 1} = \frac{1}{1} = 1$$

- $$f(x_2) = f(0.5) = \frac{1}{25(0.5)^2 + 1} = \frac{1}{25(0.25) + 1} = \frac{1}{7.25}$$

Por lo tanto, los valores de la función en estos puntos son: -

$$f(x_0) = \frac{1}{7.25}$$

-

$$f(x_1) = 1$$

-

$$f(x_2) = \frac{1}{7.25}$$

2. Definir los polinomios de Lagrange (_i(x)):

Los polinomios de Lagrange para (n = 2) son:

- $$\ell_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}$$

- $$\ell_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$$

- $$\ell_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

3. El polinomio de interpolación de Lagrange:

La aproximación de la función mediante el polinomio de Lagrange es:

$$L(x) = f(x_0) \cdot \ell_0(x) + f(x_1) \cdot \ell_1(x) + f(x_2) \cdot \ell_2(x)$$

Sustituyendo los valores de (f(x_0), f(x_1), f(x_2)) y los polinomios (_i(x)):

$$L(x) = \frac{1}{7.25} \cdot \ell_0(x) + 1 \cdot \ell_1(x) + \frac{1}{7.25} \cdot \ell_2(x)$$

Donde los polinomios

$$\ell_0(x), \ell_1(x), \ell_2(x)$$

son los expresados anteriormente.

4. Evaluación de la interpolación:

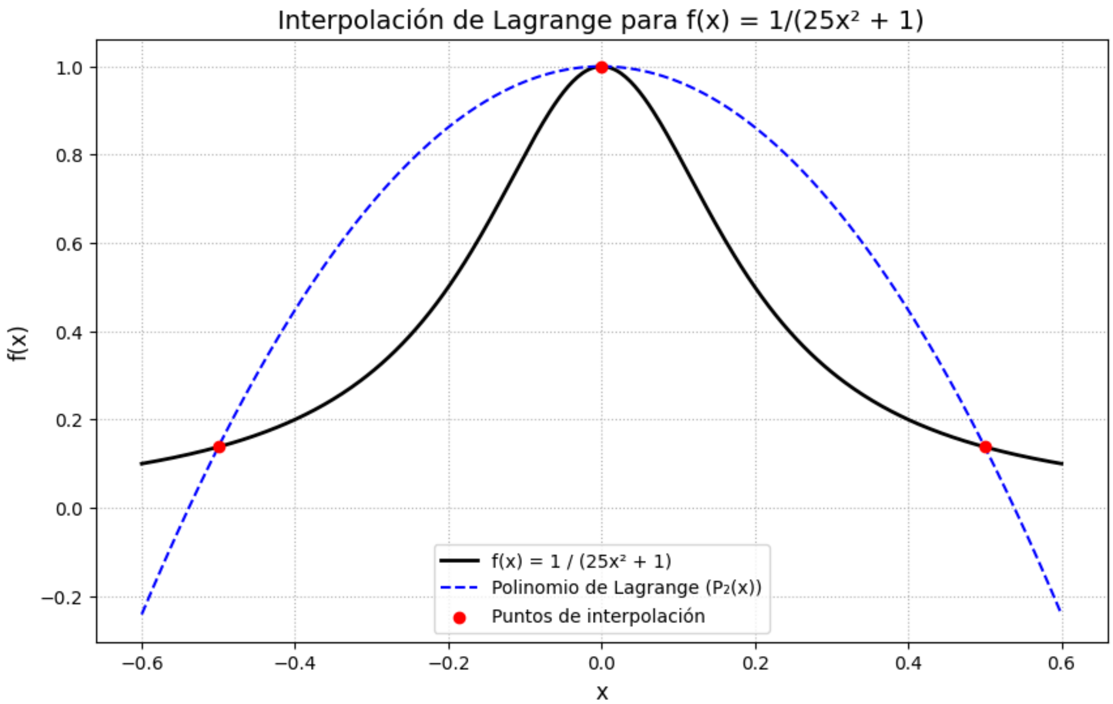
Al calcular el valor de L(x) para diferentes valores de x en el intervalo [-0.5, 0.5], obtendremos una aproximación de la función

$$f(x) = \frac{1}{25x^2 + 1}$$

usando interpolación de Lagrange.

[12]:

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return 1 / (25 * x**2 + 1)

x0 = -0.5
x1 = 0
x2 = 0.5

y0 = f(x0)
y1 = f(x1)
y2 = f(x2)

def L0(x):
    return ((x - x1) * (x - x2)) / ((x0 - x1) * (x0 - x2))

def L1(x):
    return ((x - x0) * (x - x2)) / ((x1 - x0) * (x1 - x2))

def L2(x):
    return ((x - x0) * (x - x1)) / ((x2 - x0) * (x2 - x1))

def Lagrange(x):
    return y0 * L0(x) + y1 * L1(x) + y2 * L2(x)

x_vals = np.linspace(-0.6, 0.6, 400)
y_exact = f(x_vals)
y_approx = Lagrange(x_vals)

plt.figure(figsize=(10, 6))
plt.plot(x_vals, y_exact, 'k', label="f(x) = 1 / (25x² + 1)", linewidth=2)
plt.plot(x_vals, y_approx, '--', color='blue', label="Polinomio de Lagrange␣
 ↪(P (x))")

plt.scatter([x0, x1, x2], [y0, y1, y2], color='red', zorder=5, label='Puntos de␣
 ↪interpolación')

plt.title("Interpolación de Lagrange para f(x) = 1/(25x² + 1)", fontsize=14)
plt.xlabel("x", fontsize=12)
plt.ylabel("f(x)", fontsize=12)
plt.grid(True, linestyle=':')
plt.legend()

plt.show()
```



0.3 Ejercicio 2: Serie de Taylor

Función:

$$f(x) = \arctan(x)$$

Punto de expansión:

$$x_0 = 1$$

- **Orden 0**:

$$P_0(x) = f(1) = \arctan(1) = \frac{\pi}{4}$$

- **Orden 1**: Derivada primera:

$$f'(x) = \frac{1}{1+x^2}$$

Evaluada en

$$x_0 = 1$$

:

$$f'(1) = \frac{1}{1+1^2} = \frac{1}{2}$$

Polinomio de orden 1:

$$P_1(x) = \frac{\pi}{4} + \frac{1}{2}(x-1)$$

- **Orden 2**: Derivada segunda:
$$f''(x) = -\frac{2x}{(1+x^2)^2}$$
Evaluada en
$$x_0 = 1$$
:
$$f''(1) = -\frac{2}{(1+1^2)^2} = -\frac{1}{2}$$
Polinomio de orden 2:
$$P_2(x) = \frac{\pi}{4} + \frac{1}{2}(x-1) - \frac{1}{4}(x-1)^2$$

- **Orden 3**: Derivada tercera:
$$f^{(3)}(x) = \frac{6x^2-2}{(1+x^2)^3}$$
Evaluación en
$$x = 1$$
:
$$f^{(3)}(1) = \frac{6(1)^2-2}{(1+1^2)^3} = \frac{4}{8} = \frac{1}{2}$$
Polinomio de orden 3:
$$P_3(x) = \frac{\pi}{4} + \frac{1}{2}(x-1) - \frac{1}{4}(x-1)^2 + \frac{1}{2}(x-1)^3$$

- **Orden 4**: Derivada cuarta:
$$f^{(4)}(x) = \frac{-24x(x^2-1)}{(1+x^2)^4}$$
Evaluación en
$$x = 1$$
:
$$f^{(4)}(1) = \frac{-24(1)(1^2-1)}{(1+1^2)^4} = 0$$
Polinomio de orden 4:
$$P_4(x) = \frac{\pi}{4} + \frac{1}{2}(x-1) - \frac{1}{4}(x-1)^2 + \frac{1}{2}(x-1)^3 + 0\cdot(x-1)^4$$

- **Orden 5**: Derivada quinta:
$$f^{(5)}(x) = \frac{-24(x^4-2x^2+1)}{(1+x^2)^5}$$
Evaluación en
$$x = 1$$

:

$$f^{(5)}(1) = \frac{-24(1^4 - 2(1)^2 + 1)}{(1 + 1^2)^5} = 0$$

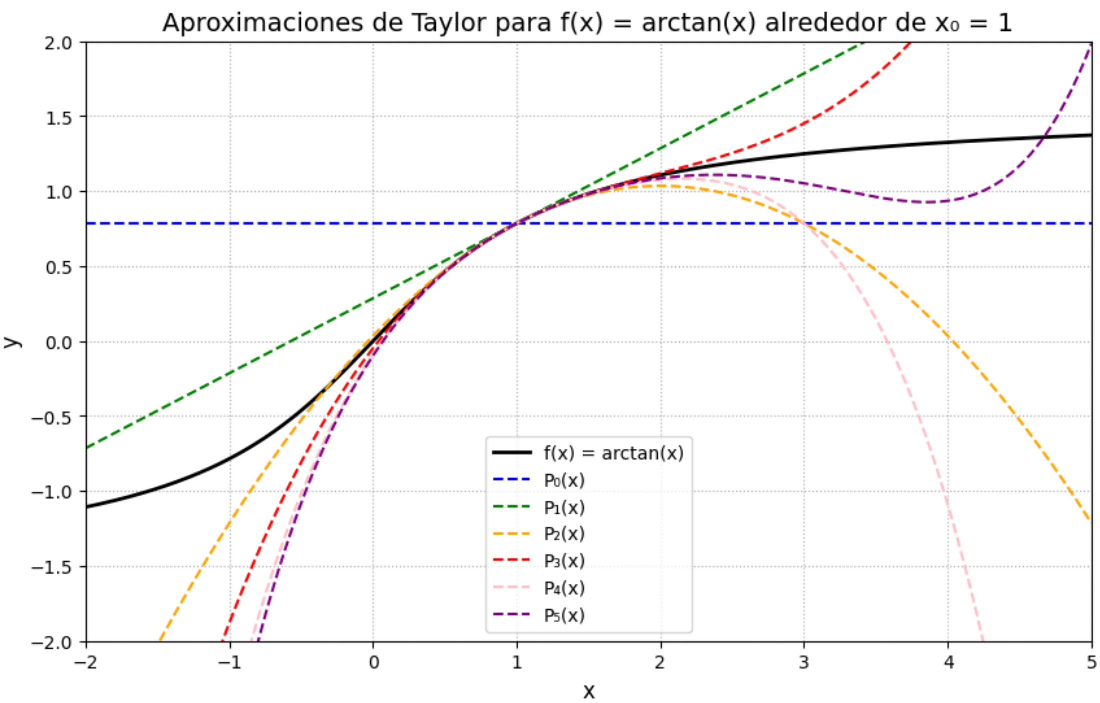
Polinomio de orden 5:

$$P_5(x) = \frac{\pi}{4} + \frac{1}{2}(x-1) - \frac{1}{4}(x-1)^2 + \frac{1}{2}(x-1)^3 + 0 \cdot (x-1)^4 + 0 \cdot (x-1)^5$$

[10]:

```
import numpy as np
import matplotlib.pyplot as plt

x_min = -2
x_max = 5
num_puntos = 600
x0 = 1

def f2(x):
    return np.arctan(x)

def P0(x):
    return np.ones_like(x) * np.pi / 4

def P1(x):
    return np.pi / 4 + (1/2) * (x - 1)

def P2(x):
    return np.pi / 4 + (1/2) * (x - 1) - (1/4) * (x - 1)**2

def P3(x):
    return np.pi / 4 + (1/2) * (x - 1) - (1/4) * (x - 1)**2 + (1/12) * (x -␣
 ↪1)**3

def P4(x):
    return np.pi / 4 + (1/2) * (x - 1) - (1/4) * (x - 1)**2 + (1/12) * (x -␣
 ↪1)**3 - (1/24) * (x - 1)**4

def P5(x):
    return np.pi / 4 + (1/2) * (x - 1) - (1/4) * (x - 1)**2 + (1/12) * (x -␣
 ↪1)**3 - (1/24) * (x - 1)**4 + (1/120) * (x - 1)**5

x = np.linspace(x_min, x_max, num_puntos)

plt.figure(figsize=(10, 6))

plt.plot(x, f2(x), 'k', label='f(x) = arctan(x)', linewidth=2)

plt.plot(x, P0(x), '--', color='blue', label='P (x)')
```

```python
plt.plot(x, P1(x), '--', color='green', label='P (x)')
plt.plot(x, P2(x), '--', color='orange', label='P (x)')
plt.plot(x, P3(x), '--', color='red', label='P (x)')
plt.plot(x, P4(x), '--', color='pink', label='P (x)')
plt.plot(x, P5(x), '--', color='purple', label='P (x)')

plt.title("Aproximaciones de Taylor para f(x) = arctan(x) alrededor de x  = 1",␣
  ↪fontsize=14)
plt.xlabel("x", fontsize=12)
plt.ylabel("y", fontsize=12)
plt.grid(True, linestyle=':')
plt.legend()
plt.xlim(x_min, x_max)
plt.ylim(-2, 2)
plt.show()
```



0.4 Ejercicio2: Aproximación de la función

$$f(x) = \arctan(x)$$

mediante la Interpolación de Lagrange

Consideramos los puntos

$$x_0 = 0, x_1 = 0.5, x_2 = 1$$

como nuestros puntos de interpolación.

1. Calcular los valores de la función en los puntos (x_0, x_1, x_2):

- $$f(x_0) = f(0) = \arctan(0) = 0$$

- $$f(x_1) = f(0.5) = \arctan(0.5) \approx 0.4636$$

- $$f(x_2) = f(1) = \arctan(1) = \frac{\pi}{4} \approx 0.7854$$

Por lo tanto, los valores de la función en estos puntos son: -

$$f(x_0) = 0$$

-

$$f(x_1) \approx 0.4636$$

-

$$f(x_2) \approx 0.7854$$

2. Definir los polinomios de Lagrange (_i(x)):

Los polinomios de Lagrange para (n = 2) son:

- $$\ell_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}$$

- $$\ell_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$$

- $$\ell_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

3. El polinomio de interpolación de Lagrange:

La aproximación de la función mediante el polinomio de Lagrange es:

$$L(x) = f(x_0) \cdot \ell_0(x) + f(x_1) \cdot \ell_1(x) + f(x_2) \cdot \ell_2(x)$$

Sustituyendo los valores de (f(x_0), f(x_1), f(x_2)) y los polinomios (_i(x)):

$$L(x) = 0 \cdot \ell_0(x) + 0.4636 \cdot \ell_1(x) + 0.7854 \cdot \ell_2(x)$$

Donde los polinomios

$$\ell_0(x), \ell_1(x), \ell_2(x)$$

son los expresados anteriormente.

4. Evaluación de la interpolación:

Al calcular el valor de

$$L(x)$$

para diferentes valores de (x) en el intervalo

$$[0, 1]$$

, obtendremos una aproximación de la función

$$f(x) = \arctan(x)$$

usando interpolación de Lagrange.

[13]:
```python
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return np.arctan(x)

x0 = 0
x1 = 0.5
x2 = 1

y0 = f(x0)
y1 = f(x1)
y2 = f(x2)

def L0(x):
    return ((x - x1) * (x - x2)) / ((x0 - x1) * (x0 - x2))

def L1(x):
    return ((x - x0) * (x - x2)) / ((x1 - x0) * (x1 - x2))

def L2(x):
    return ((x - x0) * (x - x1)) / ((x2 - x0) * (x2 - x1))

def Lagrange(x):
    return y0 * L0(x) + y1 * L1(x) + y2 * L2(x)

x_vals = np.linspace(0, 1, 400)
y_exact = f(x_vals)
y_approx = Lagrange(x_vals)

plt.figure(figsize=(10, 6))
plt.plot(x_vals, y_exact, 'k', label="f(x) = arctan(x)", linewidth=2)
plt.plot(x_vals, y_approx, '--', color='blue', label="Polinomio de Lagrange␣
 ↪(P (x))")

plt.scatter([x0, x1, x2], [y0, y1, y2], color='red', zorder=5, label='Puntos de␣
 ↪interpolación')
```

```python
plt.title("Interpolación de Lagrange para f(x) = arctan(x)", fontsize=14)
plt.xlabel("x", fontsize=12)
plt.ylabel("f(x)", fontsize=12)
plt.grid(True, linestyle=':')
plt.legend()

plt.show()
```