Serie de Taylor y Polinomios de Lagrange

Anthony Contreras

Tabla de Contenidos

Serie de Taylor y Polinomios de Lagrange

Determine el orden de la mejor aproximación para las siguientes funciones, usando la Serie de Taylor y el Polinomio de Lagrange:

125∗x2+1,x0=0

arctanx,x0=1

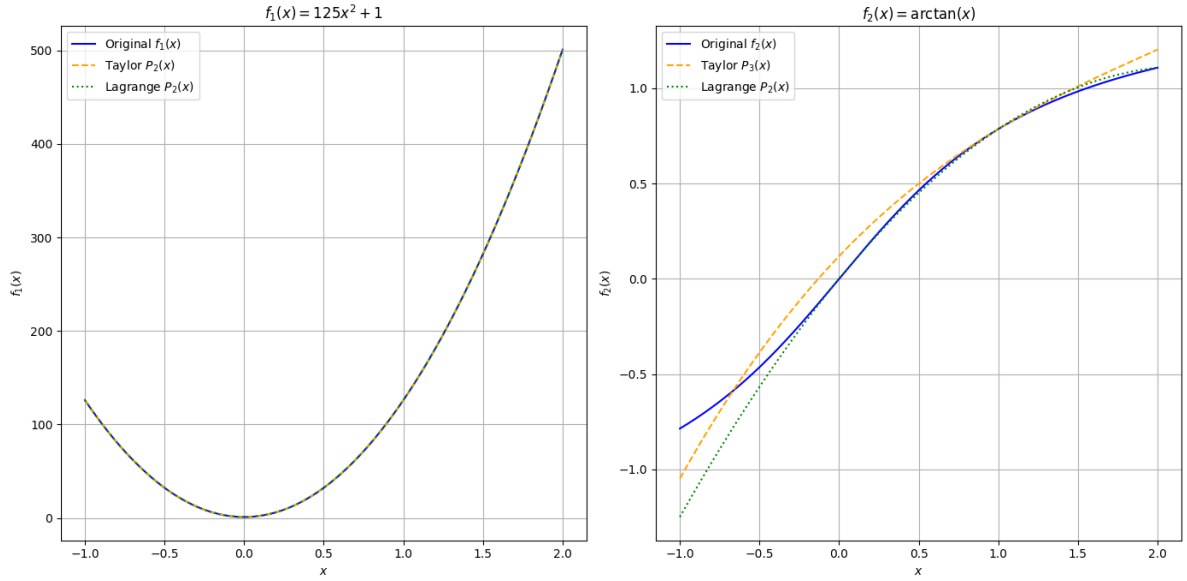
```
Escriba las fórmulas de los diferentes polinomios
Grafique las diferentes aproximaciones
```

Link Github Serie de Taylor y Polinomios de Lagrange

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

# Función f1(x) = 125x^2 + 1
def f1(x):
    return 125 * x**2 + 1

# Aproximación de Taylor para f1 en x0 = 0
def taylor_f1(x):
    # En este caso, el polinomio de Taylor coincide con la función original
    return 125 * x**2 + 1
```

```
# Función f2(x) = arctan(x)
def f2(x):
    return np.arctan(x)

# Aproximación de Taylor para f2 en x0 = 1
def taylor_f2(x):
    # Polinomio de Taylor de orden 3 para f2(x)
    return (np.pi / 4) + (1/2)*(x - 1) - (1/8)*(x - 1)**2 + (1/24)*(x - 1)**3


# Polinomio de Lagrange para f1(x)
def lagrange_f1(x):
    # Puntos de interpolación
    x0, x1, x2 = 0, 1, 2
    y0, y1, y2 = f1(x0), f1(x1), f1(x2)

    # Fórmula del Polinomio de Lagrange
    term0 = y0 * ((x - x1) * (x - x2)) / ((x0 - x1) * (x0 - x2))
    term1 = y1 * ((x - x0) * (x - x2)) / ((x1 - x0) * (x1 - x2))
    term2 = y2 * ((x - x0) * (x - x1)) / ((x2 - x0) * (x2 - x1))

    return term0 + term1 + term2

# Polinomio de Lagrange para f2(x)
def lagrange_f2(x):
    # Puntos de interpolación
    x0, x1, x2 = 0, 1, 2
    y0, y1, y2 = f2(x0), f2(x1), f2(x2)

    # Fórmula del Polinomio de Lagrange
    term0 = y0 * ((x - x1) * (x - x2)) / ((x0 - x1) * (x0 - x2))
    term1 = y1 * ((x - x0) * (x - x2)) / ((x1 - x0) * (x1 - x2))
    term2 = y2 * ((x - x0) * (x - x1)) / ((x2 - x0) * (x2 - x1))

    return term0 + term1 + term2

# ------------------------------------------------------
# Configuración del rango de x para las gráficas
x = np.linspace(-1, 2, 500)  # 500 puntos entre -1 y 2

# Creamos las gráficas
plt.figure(figsize=(14, 7))
```

```
# Gráfica para f1(x) y sus aproximaciones
plt.subplot(1, 2, 1)  # Primer subplot
plt.plot(x, f1(x), label="Original $f_1(x)$", color="blue")
plt.plot(x, taylor_f1(x), label="Taylor $P_2(x)$", color="orange", linestyle='--')
plt.plot(x, lagrange_f1(x), label="Lagrange $P_2(x)$", color="green", linestyle=':')
plt.title("$f_1(x) = 125x^2 + 1$")
plt.xlabel("$x$")
plt.ylabel("$f_1(x)$")
plt.legend()
plt.grid()

# Gráfica para f2(x) y sus aproximaciones
plt.subplot(1, 2, 2)  # Segundo subplot
plt.plot(x, f2(x), label="Original $f_2(x)$", color="blue")
plt.plot(x, taylor_f2(x), label="Taylor $P_3(x)$", color="orange", linestyle='--')
plt.plot(x, lagrange_f2(x), label="Lagrange $P_2(x)$", color="green", linestyle=':')
plt.title("$f_2(x) = \\arctan(x)$")
plt.xlabel("$x$")
plt.ylabel("$f_2(x)$")
plt.legend()
plt.grid()

# Mostramos las gráficas
plt.tight_layout()
plt.show()
```


$f_1(x) = 125x^2 + 1$
Original $f_1(x)$
Taylor $P_2(x)$
Lagrange $P_2(x)$
$f_1(x)$
x
$f_2(x) = \arctan(x)$
Original $f_2(x)$
Taylor $P_3(x)$
Lagrange $P_2(x)$
$f_2(x)$
x