HW3 E94116198 劉羿伶

1. $P_n(x)=\sum_{i=0}^{n} f(x_i)\,L_i(x)$, $L_i(x)=\prod_{j\neq i}^{n}\frac{x-x_i}{x_i-x_j}$

Error Bound : $E_n(x)=|f(x)-P_n(x)|\leq \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}|x-x_i|$

(1) degree one: $x_1=0.733$, $x_2=0.768$

$P_1(x)=f(x_1)L_1(x)+f(x_2)L_2(x)$

$P_1(x)=0.7432\times\frac{x-0.768}{0.733-0.768}+0.7193\times\frac{x-0.733}{0.768-0.733}$

$P_1(0.750)=0.7432\times 0.51429+0.7193\times 0.48571\approx 0.73159$

$E_1(0.750)=\frac{\overset{-\cos(0.733)}{|f''(\xi)|}}{2}|(x-0.733)(x-0.768)|=\frac{0.7432}{2}|(0.75-0.733)(0.75-0.768)|=1.137\times 10^{-4}$

$\therefore$ $\underline{P_1(0.750)=0.73159\text{ , }E_1(0.750)\leq 1.137\times 10^{-4}}$ ※

(2) degree two: $x_0=0.698$, $x_1=0.733$, $x_2=0.768$

$P_2(x)=f(x_0)L_0(x)+f(x_1)L_1(x)+f(x_2)L_2(x)$

$P_2(x)=0.7661\times\frac{(x-0.733)(x-0.768)}{(0.698-0.733)(0.698-0.768)}+0.7432\times\frac{(x-0.698)(x-0.768)}{(0.733-0.698)(0.733-0.768)}+0.7193\times\frac{(x-0.698)(x-0.733)}{(0.768-0.698)(0.768-0.733)}$

$P_2(0.750)=0.7661\times(-0.1249)+0.7432\times 0.76408+0.7193\times 0.36082\approx 0.731716$

$E_2(x)=\frac{\overset{\sin(0.768)}{|f'''(\xi)|}}{6}|(x-0.698)(x-0.733)(x-0.768)|$

$E_2(0.750)=\frac{0.694698}{6}|(0.750-0.698)(0.750-0.733)(0.750-0.768)|\approx 1.842\times 10^{-6}$

$\therefore$ $\underline{P_2(0.750)=0.731716\text{ , }E_2(0.750)\leq 1.842\times 10^{-6}}$ ※

(3) degree three: $x_0=0.698$, $x_1=0.733$, $x_2=0.768$, $x_3=0.803$

$P_3(x)=f(x_0)L_0(x)+f(x_1)L_1(x)+f(x_2)L_2(x)+f(x_3)L_3(x)$

$P_3(x)=0.7661\times\frac{(x-0.733)(x-0.768)(x-0.803)}{(0.698-0.733)(0.698-0.768)(0.698-0.803)}+0.7432\times\frac{(x-0.698)(x-0.768)(x-0.803)}{(0.733-0.698)(0.733-0.768)(0.733-0.803)}$

$+0.7193\times\frac{(x-0.698)(x-0.733)(x-0.803)}{(0.768-0.698)(0.768-0.733)(0.768-0.803)}+0.6946\times\frac{(x-0.698)(x-0.733)(x-0.768)}{(0.803-0.698)(0.803-0.733)(0.803-0.768)}$

$P_3(0.750)=0.7661\times(-0.06304)+0.7432\times 0.57852+0.7193\times 0.54638+0.6946\times(-0.061854)\approx 0.731708$

$E_3(x)=\frac{\overset{\cos(0.698)}{|f^{(4)}(\xi)|}}{24}|(x-0.698)(x-0.733)(x-0.768)(x-0.803)|$

$E_3(0.750)=\frac{0.7661}{24}|(0.75-0.698)(0.75-0.733)(0.75-0.768)(0.75-0.803)|=2.692\times 10^{-8}$

$\therefore$ $\underline{P_3(0.750)\approx 0.731708\text{ , }E_3(0.750)=2.692\times 10^{-8}}$ ※

(4) degree 4：只有4個已知點，最高只能做 degree 3

$\Rightarrow$ degree 4 插值 = degree 3 插值

$\underline{E_4(0.750)=E_3(0.750)\leq 2.692\times 10^{-8}}$ ※

2. $y=e^{-x}$，求 x 使 $x=y$

$x=y$	0.3	0.4	0.5	0.6
e^{-x}	0.740818	0.670320	0.606531	0.548812

$\Rightarrow 0.5 < x < 0.6$

$f(x) = x - e^{-x}$, $f'(x) = 1 + e^{-x}$

Newton's method

$$x_1 = 0.5 - \frac{0.5 - 0.606531}{1+0.606531} = 0.566311 \quad , \quad f(0.566311) = -0.0013045$$

$$x_2 = 0.566311 - \frac{f(x_1)}{f'(x_1)} = 0.566311 - \frac{-0.0013045}{1.5676155} = 0.567145 \,, \quad f(0.567145) = -4.551138 \times 10^{-7} \approx 0$$

$\therefore$ $x = 0.567145$ is an approximation to the solution $x - e^{-x} = 0$ #

3.

```python
import numpy as np
from scipy.interpolate import CubicHermiteSpline
from scipy.optimize import minimize_scalar, root_scalar

# 原始數據
T = np.array([0, 3, 5, 8, 13])  # 時間 (s)
D = np.array([0, 200, 375, 620, 990])  # 距離 (ft)
V = np.array([75, 77, 80, 74, 72])  # 速度 (ft/s)

# 建立 Hermite 插值函數
hermite_poly = CubicHermiteSpline(T, D, V)

# (a) 計算 t = 10 時的距離和速度
t_pred = 10
d_pred = hermite_poly(t_pred)  # 位置
v_pred = hermite_poly.derivative()(t_pred)  # 速度

print("===問題(a)===")
print(f"At t = {t_pred}s: Predicted Position = {d_pred:.2f} ft, Speed = {v_pred:.2f} ft/s")
# (b) 檢查何時速度超過 55 mi/h (80.67 ft/s)
def speed_exceeds_threshold(t):
    return hermite_poly.derivative()(t) - 80.67

# 設定較小的區間以確保求解成功
try:
    exceed_time = root_scalar(speed_exceeds_threshold, bracket=[3, 3.5], method='brentq').root
    print("\n===問題(b)===")
    print(f"The car first exceeds 55 mi/h at t = {exceed_time:.2f} s")
except ValueError:
    print("\n===問題(b)===")
    print("Error: Unable to find the time when the speed exceeds 55 mi/h")

# (c) 預測最大速度
result = minimize_scalar(lambda t: -hermite_poly.derivative()(t), bounds=(T[0], T[-1]), method='bounded')
max_speed = -result.fun
max_speed_time = result.x
print("\n===問題(c)===")
print(f"Predicted Maximum Speed = {max_speed:.2f} ft/s at t = {max_speed_time:.2f} s")
```

```
===問題(a)===
At t = 10s: Predicted Position = 768.96 ft, Speed = 74.64 ft/s

===問題(b)===
The car first exceeds 55 mi/h at t = 3.14 s

===問題(c)===
Predicted Maximum Speed = 92.04 ft/s at t = 4.06 s
```

#