NM hw4 report

P1.

a. Divided difference table

0	0.15000	0.176091	2.435467	-5.750502
1	0.21000	0.322219	1.975427	-3.908814
2	0.23000	0.361728	1.740898	-2.946377
3	0.27000	0.431364	1.475724	-2.230693
4	0.32000	0.505150	1.297269	
5	0.35000	0.544068		

b. For each 3 points, calculate with formula:

$$P_2'(x_i) = f[x_i, x_{i+1}] + f[x_i, x_{i+1}, x_{i+2}][(x - x_i) + (x - x_{i+1})]$$

```python
for i in range(n-2):
    xs = data[i:i+3]
    ys = [f(x) for x in xs]
    # 差商表
    t = [[0 for _ in range(3)] for _ in range(3)]
    for k in range(3):
        t[k][0] = ys[k]
    for j in range(1, 3):
        for k in range(3-j):
            t[k][j] = (t[k+1][j-1] - t[k][j-1]) / (xs[k+j] - xs[k])
```

c. Result

```
連續三點組合、插值導數、真值、誤差：
[0.15, 0.21, 0.23]  f'=1.423379  真值=1.620502  誤差=1.97e-01
[0.21, 0.23, 0.27]  f'=1.600181  真值=1.620502  誤差=2.03e-02
[0.23, 0.27, 0.32]  f'=1.634829  真值=1.620502  誤差=1.43e-02
[0.27, 0.32, 0.35]  f'=1.596182  真值=1.620502  誤差=2.43e-02

最佳連續三點組合: [0.23, 0.27, 0.32]
對應插值導數: 1.634829
真值: 1.620502
最小誤差: 1.43e-02
```

⇒ 3 points with least error are [0.23, 027, 0.32]

p2.

a.Function difference table

0	0.30	0.398507	0.261302	-0.006371	-0.002190	0.000341
1	0.50	0.659809	0.254931	-0.008561	-0.001849	0.000415
2	0.70	0.914739	0.246370	-0.010410	-0.001434	0.000472
3	0.90	1.161109	0.235960	-0.011843	-0.000961	
4	1.10	1.397069	0.224117	-0.012805		
5	1.30	1.621186	0.211312			
6	1.50	1.832498				

b.對插值多項式微分去近似多項式的微分，套用funciton difference table的值去計算

```
(a) f'(0.72) 估計值： 1.2509594908521686 ，使用點： [0.7, 0.9, 1.1, 1.3]
(b) f'(1.33) 估計值： 1.078969516076625 ，使用點： [1.1, 1.3, 1.5]
(c) f'(0.50) 估計值： 1.2925483703540175 ，使用點： [0.3, 0.5, 0.7, 0.9, 1.1]
```

p3.

(1) $f''(x_0) = C_{-2}f_{-2} + C_{-1}f_{-1} + C_0 f_0 + C_1 f_1 + C_2 f_2$

$P(u) = au^4 + bu^3 + cu^2 + du + e$

① $P(u) = 1$

$f_{-2} = f_{-1} = f_0 = f_1 = f_2 = 1$

$\Rightarrow f''(x_0) = C_{-2} + C_{-1} + C_0 + C_1 + C_2 = P''(0) = 0$

② $P(u) = u$

$f_{-2} = -2h,\ f_{-1} = -h,\ f_0 = 0,\ f_1 = h,\ f_2 = 2h$

$\Rightarrow f''(x_0) = C_{-2}(-2h) + C_{-1}(-h) + C_1(h) + C_2(2h) = P''(0) = 0$

③ $P(u) = u^2$

$f_{-2} = 4h^2,\ f_{-1} = h^2,\ f_0 = 0,\ f_1 = h^2,\ f_2 = 4h^2$

$\Rightarrow f''(x_0) = C_{-2}(4h^2) + C_{-1}(h^2) + C_1 h^2 + C_2(4h^2) = P''(0) = 2$

④ $P(u) = u^3$

$f_{-2} = -8h^3,\ f_{-1} = -h^3,\ f_1 = h^3,\ f_2 = 8h^3$

$f''(x_0) = C_{-2}(-8h^3) + C_{-1}(-h^3) + C_1 h^3 + C_2(8h^3) = P''(0) = 0$

⑤ $P(u) = u^4$

$f_{-2} = 16h^4,\ f_{-1} = -h^4,\ f_1 = h^4,\ f_2 = 16h^4$

$f''(x_0) = C_{-2}(16h^4) + C_{-1}(-h^4) + C_1 h^4 + C_2 16h^4 = P''(0) = 0$

$$\Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2h & h & 0 & h & 2h \\ 4h^2 & h^2 & 0 & h^2 & 4h^2 \\ -8h^3 & -h^3 & 0 & h^3 & 8h^3 \\ 16h^4 & h^4 & 0 & h^4 & 16h^4 \end{bmatrix} \begin{bmatrix} C_{-2} \\ C_{-1} \\ C_0 \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} C_{-2} \\ C_{-1} \\ C_0 \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} \frac{-1}{12} \\ \frac{4}{3} \\ \frac{-5}{2} \\ \frac{4}{3} \\ -1 \end{bmatrix} \cdot \frac{1}{h^2}$$

$$\Rightarrow f''(x) = \frac{-f_2 + 16f_{-1} - 30f_0 + 16f_1 - f_2}{12h^2}$$

#

(2) $f'''(x)$ $\qquad h^3 \to 3\text{-}2\text{-}1 = 6$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2h & -h & 0 & h & 2h \\ 4h^2 & h^2 & 0 & h^2 & 4h^2 \\ -8h^3 & -h^3 & 0 & h^3 & 8h^3 \\ 16h^4 & h^4 & 0 & h^4 & 16h^4 \end{bmatrix} \begin{bmatrix} c_{-2} \\ c_{-1} \\ c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} c_{-2} \\ c_{-1} \\ c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ 1 \\ 0 \\ -1 \\ \frac{1}{2} \end{bmatrix} \cdot \frac{1}{h^3}$$

$$\Rightarrow f'''(x) = \frac{-f_{-2} + f_{-1} - f_1 + f_2}{2h^3}$$

#

P4

a. 全部使用 1/3 rule 作為基準

```python
# 1. 全部用 1/3 rule
I_13_all = (
    simpson_13(y[0], y[1], y[2], h) +
    simpson_13(y[2], y[3], y[4], h) +
    simpson_13(y[4], y[5], y[6], h) +
    simpson_13(y[6], y[7], y[8], h)
)
print(f"全部用 1/3 rule (正確答案): {I_13_all:.6f}")
```

```
全部用 1/3 rule (正確答案): 1.766933
```

b. 2 組 3/8 + 1 組 1/3 (4,4,3), (4,3,4), (3,3,4)

```python
# (a) 3/8 on [0:3], 3/8 on [3:6], 1/3 on [6:8]
I_a = simpson_38(y[0], y[1], y[2], y[3], h) + simpson_38(y[3], y[4], y[5], y[6], h) + simpson_13(y[6], y[7], y[8], h)
results.append(("3/8[0:3],3/8[3:6],1/3[6:8]", I_a, abs(I_a - I_13_all)))

# (b) 3/8 on [0:3], 1/3 on [3:5], 3/8 on [5:8]
I_b = simpson_38(y[0], y[1], y[2], y[3], h) + simpson_13(y[3], y[4], y[5], h) + simpson_38(y[5], y[6], y[7], y[8], h)
results.append(("3/8[0:3],1/3[3:5],3/8[5:8]", I_b, abs(I_b - I_13_all)))

# (c) 1/3 on [0:2], 3/8 on [2:5], 3/8 on [5:8]
I_c = simpson_13(y[0], y[1], y[2], h) + simpson_38(y[2], y[3], y[4], y[5], h) + simpson_38(y[5], y[6], y[7], y[8], h)
results.append(("1/3[0:2],3/8[2:5],3/8[5:8]", I_c, abs(I_c - I_13_all)))
```

c. 取誤差最小的一組

```
2 組 3/8 + 1 組 1/3 (連續分法) 與正確答案的誤差：
3/8[0:3],3/8[3:6],1/3[6:8]: 積分值=1.766946, 誤差=1.25e-05
3/8[0:3],1/3[3:5],3/8[5:8]: 積分值=1.766950, 誤差=1.67e-05
1/3[0:2],3/8[2:5],3/8[5:8]: 積分值=1.766946, 誤差=1.25e-05

最佳分法: 3/8[0:3],3/8[3:6],1/3[6:8], 誤差=1.25e-05
```

⇒ 3/8 rule on [1,4], [4,7] and 1/3 rule on [7,9]

P5

a. use trapezoidal rule with halving h on every iteration until the integral value does not differ more than 0.02

```python
while True:
    n *= 2
    I_new = trapezoidal(f_trap, a, b, n)
    print(f"n={n:4d}, h={(b-a)/n:.5f}, integral={I_new:.6f}, diff={abs(I_new-I_old):.6f}")
    if abs(I_new - I_old) < 0.02:
        break
    I_old = I_new
h_final = (b - a) / n
```

b. Result

```
Adaptive Trapezoidal Rule for f(x) = 1/x^2 over [0.2, 1]:
n=   2, h=0.40000, integral=6.311111, diff=4.088889
n=   4, h=0.20000, integral=4.718056, diff=1.593056
n=   8, h=0.10000, integral=4.197677, diff=0.520378
n=  16, h=0.05000, integral=4.051043, diff=0.146635
n=  32, h=0.02500, integral=4.012876, diff=0.038166
n=  64, h=0.01250, integral=4.003227, diff=0.009650

Terminated at h = 0.01250, integral = 4.003227
```

⇒ h = 0.0125, integral = 4.003227

P6

先積y再積x，每次以panel width = h (0.1)去積分，並將[a,b]映射成[-1,1]再加上講義中給的3-term高斯節點以及權重，積分出面積。

```python
# 三點高斯積分的節點與權重
nodes = np.array([-0.77459667,0,0.77459667])
weights = np.array([0.55555555,0.88888889,0.55555555])

# 區間設定與步長 h
x_start, x_end = -0.2, 1.4
y_start, y_end = 0.4, 2.6
h = 0.1
  Ctrl+L to chat, Ctrl+K to generate
# 累加積分值
integral = 0.0
x_panels = np.arange(x_start, x_end, h)
y_panels = np.arange(y_start, y_end, h)

for i in range(len(x_panels)):
    for j in range(len(y_panels)):
        x_a, x_b = x_panels[i], x_panels[i] + h
        y_a, y_b = y_panels[j], y_panels[j] + h

        for xi in range(3):
            for yi in range(3):
                # 範圍映射到 [-1,1] -> [a,b]
                x = 0.5 * (x_b - x_a) * nodes[xi] + 0.5 * (x_b + x_a)
                y = 0.5 * (y_b - y_a) * nodes[yi] + 0.5 * (y_b + y_a)
                w = weights[xi] * weights[yi]
                integral += w * f(x, y) * 0.25 * (x_b - x_a) * (y_b - y_a)
```

P7

$$f_{even}(x) = f(|x|)$$

$$\Rightarrow f_{even}(x) = e^{-x}\sin(2x-1),\ x \in [0,2]$$

$$f_{even}(x) = e^{x}\sin(-2x-1),\ x \in [-2,0]$$

$$2L = 4 \Rightarrow L = 2$$

$$a_n = \frac{1}{L}\int_{-L}^{L} f_{even}(x)\cos\left(\frac{n\pi x}{L}\right)dx$$

$$= \frac{2}{L}\int_{0}^{L} f_{even}(x)\cos\left(\frac{n\pi x}{L}\right)dx$$

$$= \frac{2}{2}\int_{0}^{2} f_{even}(x)\cos\left(\frac{n\pi x}{2}\right)dx$$

$$\Rightarrow a_n = \int_{0}^{2} f(x)\cos\left(\frac{n\pi x}{2}\right)dx$$

#