

Student name: Abao Zhang (张宝)

Student number: 12332459

## Homework 4

### Question 1

(a)

可以使用数学归纳法证明:

- 当  $k = 1$  时,  $f^{(1)} = \frac{1}{1+x} = (-1)^{1-1} \frac{(1-1)!}{(1+x)^1}$  成立,
- 假设  $k = i$  时成立, 则有:

$$f^{(i)}(x) = (-1)^{i-1} \frac{(i-1)!}{(1+x)^i}$$

而  $k = i + 1$  时

$$\begin{aligned} f^{(i+1)}(x) &= \frac{d(f^{(i)}(x))}{dx} = (-1)^{i-1} \frac{(i-1)!}{(1+x)^{i+1}} (-i) \\ &= (-1)^i \frac{i!}{(1+x)^{i+1}} \end{aligned}$$

故  $k = i + 1$  时也成立。故  $f^{(k)} = (-1)^{k-1} \frac{(k-1)!}{(1+x)^k}$  成立, 命题得证。

(b)

根据泰勒展开式可知:

$$P_{N(x)} = \sum_{k=0}^N \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

当  $x_0 = 0$ , 代入  $f^{(k)}(0) = (-1)^{k-1} ((k-1)!) \neq 0$  和  $f(0) = 0$ :

$$P_{N(x)} = \sum_{k=0}^N \frac{(-1)^{k-1}}{k} x^k = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + \frac{(-1)^{N-1} x^N}{N}$$

## Question 2

(a)

$$P(4) = -0.02 \cdot 4^3 + 0.1 \cdot 4^2 - 0.2 \cdot 4 + 1.66 = 1.18$$

(b)

$$P'(x) = -0.06x^2 + 0.2x - 0.2$$

代入  $x = 4$ , 求得  $P'(4) = -0.36$

(c)

$$\begin{aligned}\int_1^4 P(x)dx &= \int_1^4 (-0.02x^3 + 0.1x^2 - 0.2x + 1.66)dx \\ &= \left( -0.005x^4 + \frac{0.1x^3}{3} - 0.1x^2 + 1.66x \right) \Big|_1^4 \\ &= 6.855\end{aligned}$$

(d)

$$P(5.5) = -0.02 \cdot 5.5^3 + 0.1 \cdot 5.5^2 - 0.2 \cdot 5.5 + 1.66 = 0.66$$

(e)

不妨假设  $P(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ , 代入四个点, 即可组成关于  $a_i, i \in \{0, 1, 2, 3\}$  的方程组, 简写为  $AX = B$ ,  $A$  为范德蒙行列式, 必有唯一解。可以直接通过解方程组的方式解出对应系数。

### Question 3

(a)

首先计算三个插值点的值

$$y_0 = f(x_0) = f(1) = 1$$

$$y_1 = f(x_1) = f(1.25) \approx 1.32$$

$$y_2 = f(x_2) = f(1.5) \approx 1.84$$

由拉格朗日插值法：

$$P_{n(x)} = \sum_{i=0}^n y_i \prod_{i=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

代入三个插值点

$$\begin{aligned} P_2(x) &= y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{(x - 1.25)(x - 1.5)}{(-0.25) \cdot (-0.5)} + y_1 \frac{(x - 1)(x - 1.5)}{0.25 \cdot (-0.25)} + y_2 \frac{(x - 1)(x - 1.25)}{0.5 \cdot 0.25} \\ &= 1.55x^2 - 2.20x + 1.65 \end{aligned}$$

上面相关系数已经四舍五入保留了两位小数，这部分手动计算（分配律）比较复杂，可以使用我编写的脚本 question-3.py 进行验证。

(b)

使用  $P_2(x)$  作为  $f(x)$  估计计算  $[1, 1.5]$  上的均值为：

$$\frac{\int_1^{1.5} P_2(x) dx}{1.5 - 1} \approx 1.35$$

这里使用代码进行计算了，具体如下

```
def IP(x):  
    return (1.54951318788779 * x**3 / 3 - 2.1995483555447 * x**2 / 2 +  
    1.65003516765691 * x)  
print("均值:", (IP(1.5) - IP(1)) / (1.5 - 1))
```

(c)

有题意可知，此时  $h = 0.25$ ，下面计算  $M_3$

$$|f^{(N+1)}(x)| \leq M_{N+1}$$

$$M_3 = \max |f^{(3)}(x)| = \max \frac{d(x^x)}{dx} = \max x^x \left( (\ln(x) + 1)^3 + 3 \frac{\ln(x) + 1}{x} - \frac{1}{x^2} \right)$$

根据  $f^x$  性质可以知道，其在  $x = 1.5$  上取得极值，故  $M_3 \approx 9.45$  最后计算误差

$$|E_2(x)| \leq \frac{h^3 M_3}{9\sqrt{3}} \approx 0.01$$

使用 question-3.py 计算，可以得到  $E_2(x) = 0.009469975837730113$  这一较精确的值

## Question 4

对于本题，简单代入三个点的值组成方程组即可，本质是一个解关于 $A, B, C$ 方程组，这里使用question-4.py 进行计算

(a)

代入几个值

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 9 \end{bmatrix}$$

$$\text{解得} \begin{cases} A=3 \\ B=-2 \\ C=4 \end{cases}$$

(b)

代入几个值

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 2.5 \\ 0 \\ 4 \end{bmatrix}$$

$$\text{解得} \begin{cases} A=3.5 \\ B=-2.5 \\ C=1.5 \end{cases}$$

(c)

代入几个值

$$\begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 & 3 \\ 1 & 3 & 2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 2.5 \\ 0 \\ 4 \end{bmatrix}$$

$$\text{解得} \begin{cases} A \approx 7.33 \\ B \approx -1.33 \\ C \approx 0.33 \end{cases}, \text{结果保留了两位小数}$$

(d)

不能，因为线性方程组无解，也就是 $\begin{bmatrix} 1 & 1 & 2 \\ 1 & 3 & 2 \\ 1 & 1 & 2 \end{bmatrix}$ 的行列式值为0。故无法找到对应的 $A, B, C$

## Question 5

根据牛顿插值多向式编写脚本，使用 question-5.py 进行计算，下面给出代码：

```
import numpy as np
from functools import reduce
import sympy as sp

def f(x):
    return np.exp(-x)

def newton(f, x):
    N = len(x)
    fx = f(x)
    print(fx)
    next_fx = fx
    f_f = []
    for i in range(2, N + 1):
        next_fx = np.array(
            [
                (next_fx[j + 1] - next_fx[j]) / (x[j + i - 1] - x[j])
                for j in range(N - i + 1)
            ]
        )
        f_f.append(next_fx)
        print(next_fx)

    X = sp.Symbol("x")
    Pi = fx[0]
    for i in range(N - 1):
        Pi = Pi + f_f[i][0] * reduce(
            lambda a, b: a * b, [(X - xi) for xi in x[: i + 1]]
        )
        print(f"P_{i+1}=", Pi.expand())

newton(f, np.array([0, 1, 2, 3, 4]))
print("增加 x=0.5, 1.5 之后")
newton(f, np.array([0, 1, 2, 3, 4, 0.5, 1.5]))
```

(a)

$k$	$x_k$	$f[x_k]$	First divided difference	Second divided difference	Third divided difference	Fourth divided difference
0	0	1.				
1	1	0.36787944	-0.63212056			
2	2	0.13533528	-0.23254416	0.1997882		
3	3	0.04978707	-0.08554821	0.07349797	-0.04209674	
4	4	0.01831564	-0.03147143	0.02703839	-0.01548653	0.00665255

(b)

$$\begin{aligned}P_1(x) &= f(x_0) + f[x_0, x_1](x - x_0) = 1 - 0.63212056x \\P_2(x) &= P_1(x) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\&= 0.199788200446864x^2 - 0.831908759275422x + 1.0 \\P_3(x) &= -0.0420967429712745x^3 + 0.326078429360688x^2 - \\&\quad 0.916102245217971x + 1.0 \\P_4(x) &= 0.00665255417296605x^4 - 0.0820120680090708x^3 + \\&\quad 0.399256525263314x^2 - 0.956017570255767x + 1.0\end{aligned}$$

(c)

增加  $x = 0.5, 1.5$  两个采样点后，可以额外计算  $P_5, P_6$

$$\begin{aligned}P_5(x) &= -0.00165758860494877x^5 + 0.0232284402224537x^4 - \\&\quad 0.140027669182278x^3 + 0.482135955510753x^2 - \\&\quad 0.995799696774538x + 1.0 \\P_6(x) &= 0.000276960893235427x^6 - 0.00456567798392075x^5 + \\&\quad 0.0343068759518708x^4 - 0.158722529475669x^3 + \\&\quad 0.495707039279288x^2 - 0.999123227493363x + 1.0\end{aligned}$$

(d)

根据泰展展开公式  $f(x) = P_{n(x)} + R_n(x)$  可以知道：

$$f(x) - P_6(x) = R_6(x) = \frac{f^{(7)}(c)}{7!}x^7 = -\frac{e^{-c}}{7!}x^7$$

其中  $R_6(x)$  为拉格朗日余项， $c \in (-\infty, \infty)$ ，也就是说  $\begin{cases} f(x) > P_6(x) & \text{if } x < 0 \\ f(x) = P_6(x) & \text{if } x = 0 \\ f(x) < P_6(x) & \text{if } x > 0 \end{cases}$

## Question 6

本文编写了 question-6.py 与前一问采用相同的 newton(f, x) 函数, 容易得到:

$$P_2(x) = -0.0696792757761424x^2 - 0.386739660114694x + 1.0$$

其多项式误差  $E_2(x)$  表示为如下, 其中  $c, x \in [0, \pi]$

$$\begin{aligned} |E_2(x)| &= \left| \frac{f^{(3)}(c)}{3!} \prod_{i=0}^2 (x - x_i) \right| \\ &= \left| x^3 \frac{\sin(c)}{6} - 0.785398163397448x^2 \sin(c) + 0.822467033424113x \sin(c) \right| \end{aligned}$$

提取  $\sin(c)$  可以发现

$$\begin{aligned} |E_2(x)| &\leq \left| \frac{x^3}{6} - 0.785398163397448x^2 + 0.822467033424113x \right| \\ &\leq 0.248631697054710 \end{aligned}$$

在计算中, 我使用了如下代码

```
def f(x):
    if isinstance(x, sp.Symbol):
        return sp.cos(x)
    return np.cos(math.pi * x)

def E(f, X):
    N = len(X)
    x = sp.Symbol("x")
    c = sp.Symbol("c")
    Y = f(X)

    Enx = (
        reduce(lambda a, b: a * b, [x - xi for xi in X])
        * sp.diff(f(c), c, N)
        / math.factorial(N)
    )
    print(f"E_{N-1}(x)=", Enx.expand())
    print(f"E_{N-1}(x)<=", sp.maximum(Enx.subs(c, np.pi / 2), x, sp.Interval(0,
np.pi)))

newton(f, np.array([0, np.pi / 2, np.pi]))
E(f, np.array([0, np.pi / 2, np.pi]))
```

可以得到如下输出:

```
$ python homework4/question-6.py
[ 1.          0.22058404 -0.90268536]
[-0.49619161 -0.71509551]
[-0.06967928]
P_1= 1.0 - 0.496191610557587*x
P_2= -0.0696792757761424*x**2 - 0.386739660114694*x + 1.0
E_2(x)= x**3*sin(c)/6 - 0.785398163397448*x**2*sin(c) + 0.822467033424113*x*sin(c)
E_2(x)<= 0.248631697054710
```

## Question 7

根据拉格朗日插值多项式公式

$$P_{N(x)} = \sum_{i=0}^N y_i L_{N,i}(x) = \sum_{i=0}^N y_i \prod_{j=0, j \neq i}^N \frac{x - x_j}{x_i - x_j}$$

我们可以得到

$$\begin{aligned} L_{2,0}(x) &= \prod_{j=0, j \neq 0}^2 \frac{x - x_j}{x_0 - x_j} \\ &= \frac{(x - 0)(x - \cos(\frac{\pi}{6}))}{(\cos(\frac{5\pi}{6}) - 0)(\cos(\frac{5\pi}{6}) - \cos(\frac{\pi}{6}))} \\ &= -\frac{x}{\sqrt{3}} + \frac{2x^2}{3} \\ L_{2,1}(x) &= \prod_{j=0, j \neq 1}^2 \frac{x - x_j}{x_1 - x_j} \\ &= \frac{(x - \cos(\frac{5\pi}{6}))(x - \cos(\frac{\pi}{6}))}{(0 - \cos(\frac{5\pi}{6}))(0 - \cos(\frac{\pi}{6}))} \\ &= 1 - \frac{4x^2}{3} \\ L_{2,2}(x) &= \prod_{j=0, j \neq 2}^2 \frac{x - x_j}{x_2 - x_j} \\ &= \frac{(x - \cos(\frac{5\pi}{6}))(x - 0)}{(\cos(\frac{\pi}{6}) - \cos(\frac{5\pi}{6}))(\cos(\frac{\pi}{6}) - 0)} \\ &= \frac{\sqrt{3}x}{3} + \frac{2x^2}{3} \\ &= \frac{x}{\sqrt{3}} + \frac{2x^2}{3} \end{aligned}$$



## Question 8

已知帕德近似如下：

$$R_{N,M}(x) = \frac{P_{N(x)}}{Q_{M(x)}} = \frac{\sum_{j=0}^N p_j x^j}{\left(1 + \sum_{k=1}^M q_k x^k\right)}$$

代入  $m=2, n=2$ ，于是有

$$R_{2,2}(x) = \frac{P_2(x)}{Q_2(x)} = \frac{p_0 + p_1 x + p_2 x^2}{1 + q_1 x + q_2 x^2}$$

为了计算  $R_{2,2}$ ，是可以使用  $f(x)Q_{M(x)} - P_{N(x)} = Z(x)$ ，右侧  $Z(x)$  是  $x^5$  的同阶无穷小

$$\left(1 - \frac{x}{2} + \frac{x^2}{3} - \frac{x^3}{4} + \frac{x^4}{5} - \dots\right)(1 + q_1 x + q_2 x^2) - (p_0 + p_1 x + p_2 x^2) = 0x + 0x^2 + 0x^3 + 0x^4 + c_1 x^5 + c_2 x^6 + \dots$$

根据对应系数相等可以列出方程组：

$$\begin{aligned} 1 - p_0 &= 0 \\ -p_1 + q_1 - \frac{1}{2} &= 0 \\ -p_2 - \frac{q_1}{2} + q_2 + \frac{1}{3} &= 0 \\ \frac{q_1}{3} - \frac{q_2}{2} - \frac{1}{4} &= 0 \\ -\frac{q_1}{4} + \frac{q_2}{3} + \frac{1}{5} &= 0 \end{aligned}$$

解得

$$\begin{cases} p_0 = 1 \\ p_1 = \frac{7}{10} \\ p_2 = \frac{1}{30} \\ q_1 = \frac{6}{5} \\ q_2 = \frac{3}{10} \end{cases}$$

故

$$R_{2,2}(x) = \frac{1 + \frac{7}{10}x + \frac{1}{30}x^2}{1 + \frac{6}{5}x + \frac{3}{10}x^2}$$

解方程部分可以参考 `question-8.py`

**(b)**

联立两个方程组

$$\begin{cases} f(x) = \frac{\ln(1+x)}{x} \\ f(x) \approx R_{2,2}(x) = \frac{1 + \frac{7}{10}x + \frac{1}{30}x^2}{1 + \frac{6}{5}x + \frac{3}{10}x^2} \end{cases}$$

即可得到

$$\frac{\ln(1+x)}{x} \approx \frac{1 + \frac{7}{10}x + \frac{1}{30}x^2}{1 + \frac{6}{5}x + \frac{3}{10}x^2}$$

$$\ln(1+x) \approx \frac{x(1 + \frac{7}{10}x + \frac{1}{30}x^2)}{1 + \frac{6}{5}x + \frac{3}{10}x^2} = \frac{30x + 21x^2 + x^3}{30 + 36x + 9x^3}$$

即命题成立

## Question 9

这题我使用 python 重写了算法，采用了与题目三类似的方法计算，核心代码如下

```
from functools import reduce
import numpy as np
import sympy as sp
import matplotlib.pyplot as plt

def fit(f, X):
    N = len(X)
    Y = f(X)
    x = sp.Symbol("x")
    Pn = reduce(
        lambda a, b: a + b,
        [
            Y[i]
            * reduce(
                lambda a, b: a * b,
                [(x - X[j]) / (X[i] - X[j]) for j in range(N) if j != i],
            )
            for i in range(N)
        ],
    )

    def inference(val):
        return Pn.subs(x, val)

    return np.vectorize(inference)

# input 为采样点，也就是 x_i
input = np.array([0, 0.2, 0.4, 0.6, 0.8, 1])

# plot_x 会绘制曲线的间隔点
plot_x = np.linspace(-1, 2, 10, endpoint=True)

这里使用了 sympy 库进行多项式表达，并定义了采样点，绘图仅仅绘制  $x \in [-1, 2]$  中的曲线，
并将采样点使用蓝色小点进行标识。

总体而言，拟合效果在  $x \in [-1, 2]$  中都比较不错。
```

(a)

绘制代码如下：

```
f = np.exp
plt.plot(plot_x, fit(f, input)(plot_x), label="P(x)")
plt.plot(plot_x, f(plot_x), label="f(x)")
plt.scatter(input, f(input))
plt.legend()
plt.savefig("figure-exp.png")
plt.show()
```

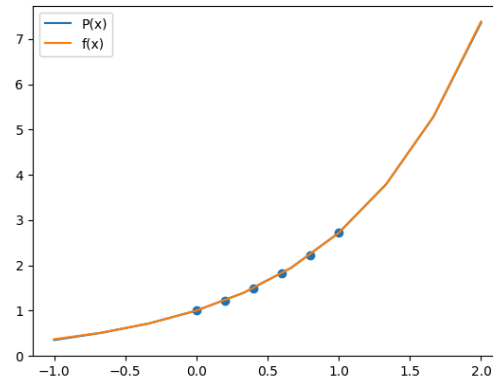


Figure 1:  $f(x) = e^x$ 及其拉格朗日近似 $P(x)$

(b)

绘制代码如下：

```
f = np.sin
plt.plot(plot_x, fit(f, input)(plot_x), label="P(x)")
plt.plot(plot_x, f(plot_x), label="f(x)")
plt.scatter(input, f(input))
plt.legend()
plt.savefig("figure-sin")
plt.show()
```

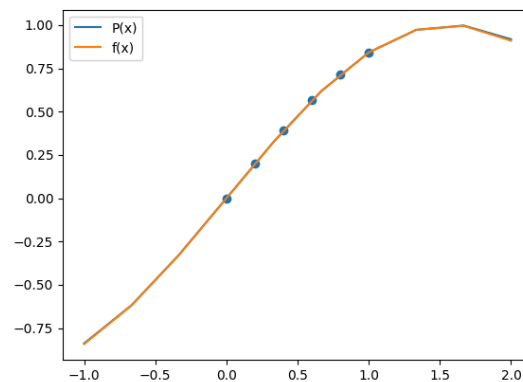


Figure 2:  $f(x) = \sin(x)$ 及其拉格朗日近似 $P(x)$

(c)

绘制代码如下：

```
f = np.vectorize(lambda x: (x + 1) ** (x + 1))
plt.plot(plot_x, fit(f, input)(plot_x), label="P(x)")
plt.plot(plot_x, f(plot_x), label="f(x)")
plt.scatter(input, f(input))
plt.legend()
plt.savefig("figure-x+1")
plt.show()
```

这里略有不同，预先定义了函数 $f(x) = (1 + x)^{1+x}$ 。

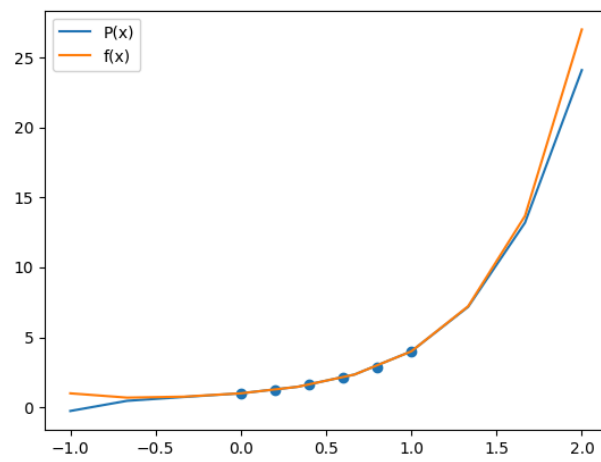


Figure 3:  $f(x) = (x + 1)^{x+1}$ 及其拉格朗日近似 $P(x)$