



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

实验成绩

实验报告

课程名称: 数值逼近

实验项目: 最佳平方逼近和最小二乘法

所在院系: 信息与计算科学

学生姓名: 葛煜龙

学生学号: 1201200206

授课学期: 22 秋

完成时间: 2022.10.29

1 习题一 最佳平方逼近多项式

下面即为绘制代码。

```
import math
import numpy as np
import matplotlib.pyplot as plt
from scipy import integrate
from scipy import linalg

def multi_xk(x, k1, k2):
    return math.pow(x, k1) * math.pow(x, k2)

def multi_f(x, k):
    return math.exp(x) * math.pow(x, k)

def multi_g(x, k):
    return abs(x) * math.pow(x, k)

def get_amatrix(k):
    amatrix = np.zeros((k + 1, k + 1))
    for row in range(0, k + 1):
        for column in range(0, k + 1):
            amatrix[row][column], wucha = integrate.quad(multi_xk, -1
                                                            , 1, args=(row, column
                                                            ))
    return amatrix

def get_bmatrix(function, k):
    bmatrix = np.zeros(k + 1)
    for column in range(0, k + 1):
        bmatrix[column], wucha = integrate.quad(function, -1, 1, args
                                                  =column)
    return bmatrix

#将函数中math.exp(i)换为abs(i)
def delta_f(x, coefficient, k):
    delta = []

    for i in x:
```

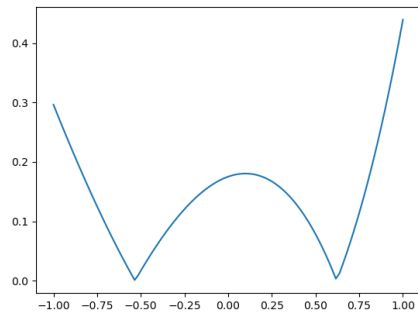
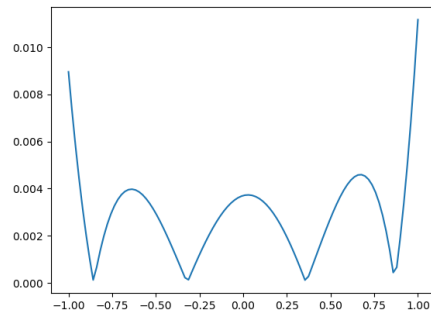
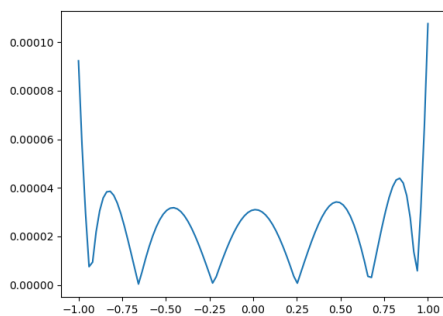
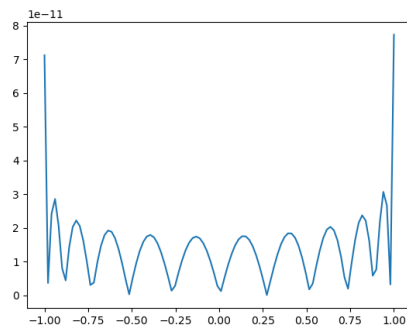
```

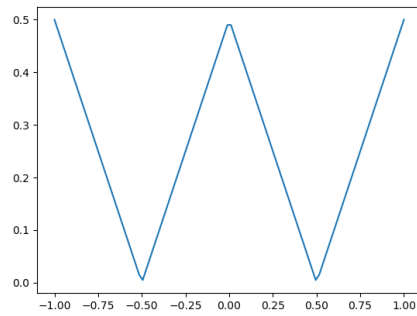
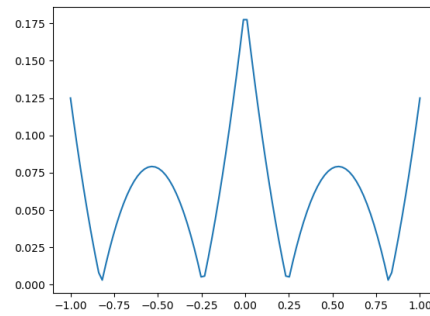
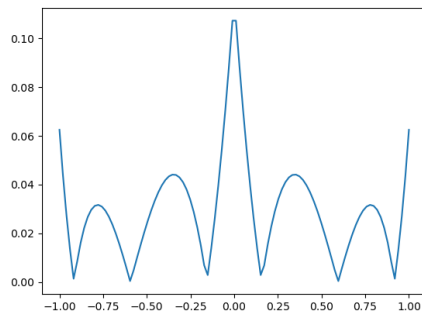
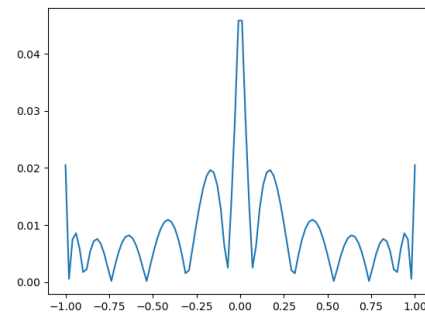
pk = np.array([])
for j in range(0, k + 1):
    pk = np.append(pk, [math.pow(i, j)])
pkplus = np.matmul(coefficent, np.transpose(pk))
delta.append(abs(abs(i) - pkplus))
return delta

if __name__ == '__main__':
    k = 10
    a = get_amatrix(k)
    b = get_bmatrix(multi_g, k)
    coefficent = linalg.solve(a, b)

    x = np.linspace(-1, 1, 100)
    delta = delta_f(x, coefficent, k)
    plt.plot(x, delta)
    plt.show()

```

(a) $k = 1$ (b) $k = 3$ (c) $k = 5$ (d) $k = 10$ 图 1: e^x 最佳平方逼近多项式

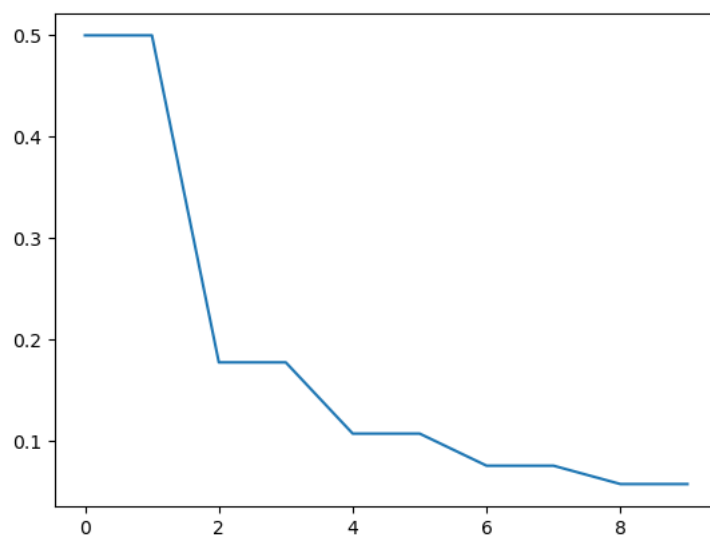
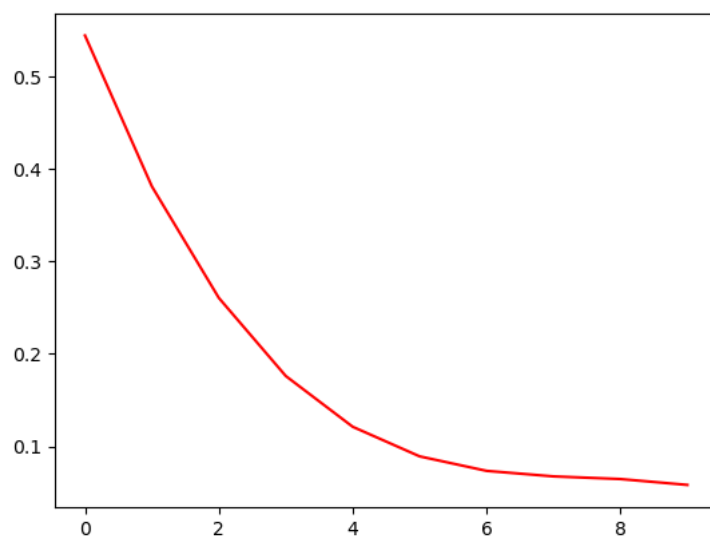
(a) $k = 1$ (b) $k = 3$ (c) $k = 5$ (d) $k = 10$ 图 2: $|x|$ 最佳平方逼近多项式

以下为 $\Delta_k \sim k$ 曲线绘制代码

```
from one import *
from scipy.optimize import curve_fit
def func(x, a, b, c, d):
    return a + b * x + c * pow(x, 2) + d * pow(x, 3)
if __name__ == '__main__':
    k = np.arange(0, 10, 1)
    deltadelta = []
    for ki in k:
        a = get_amatrix(ki)
        b = get_bmatrix(multi_g, ki)
        coefficient = linalg.solve(a, b)

        x = np.linspace(-1, 1, 100)
        delta = delta_f(x, coefficient, ki)
        deltadelta.append(max(delta))
    # 非线性最小二乘法拟合
    popt, pcov = curve_fit(func, k, deltadelta)
    # 获取popt里面是拟合系数
    a,b,c,d = popt[0],popt[1],popt[2],popt[3]
```

```
yvals = func(k, a, b, c, d) # 拟合 y 值
```

图 3: $\Delta_k \sim k$ 图 4: $\Delta_k \sim k$ 最小二乘拟合

2 习题二 正交多项式的应用

注：由于代码过长，不在页面进行展示。

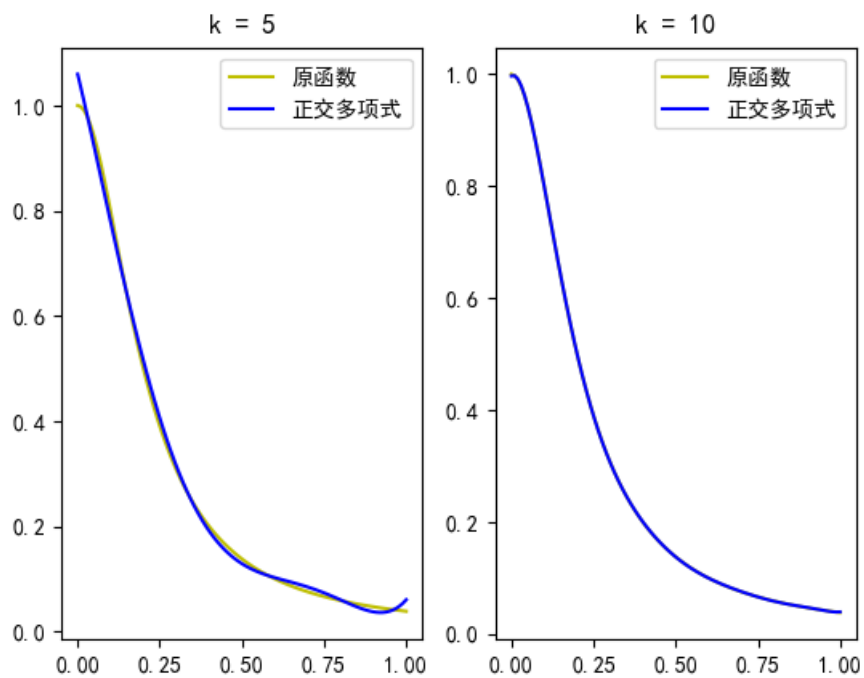
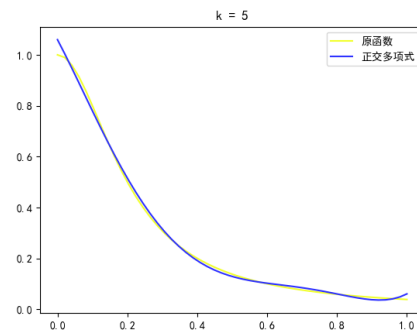
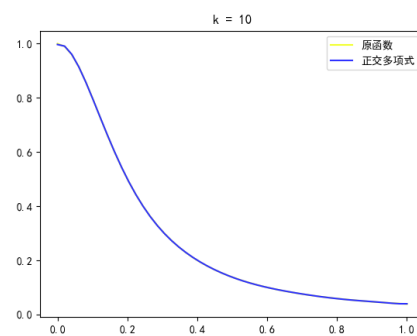
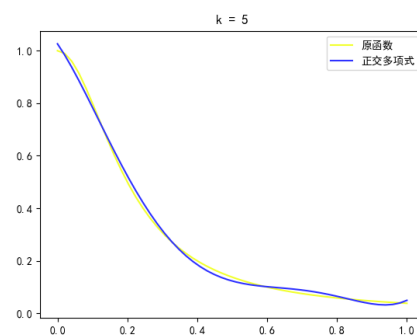
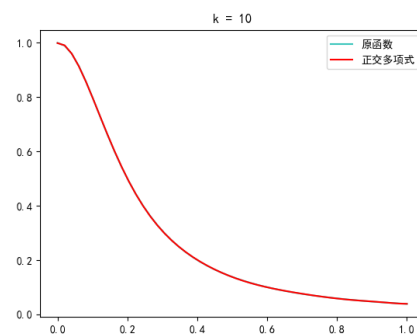


图 5: $\{x^i\}$ 为基函数

(a) $k = 5$ (b) $k = 10$ 图 6: *Legendre* 多项式为基函数(a) $k = 5$ (b) $k = 10$ 图 7: *Chebyshev* 多项式为基函数

3 习题三 多项式最小二乘法

将三个问题的图像绘制在一起如下图：

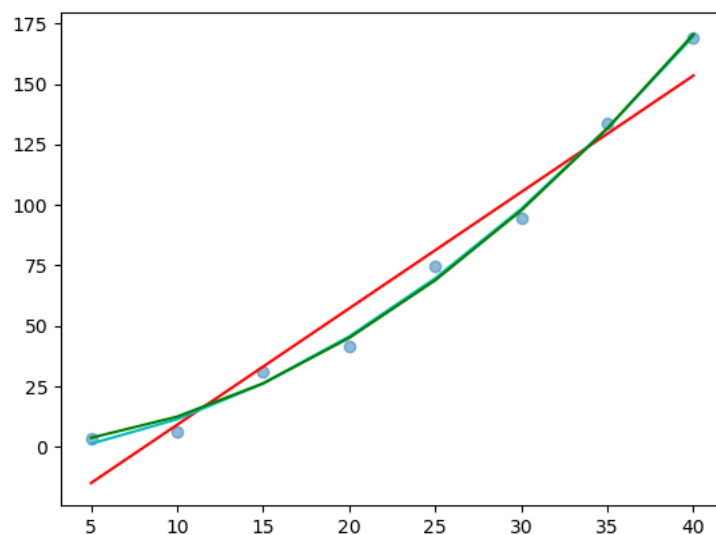


图 8: 多项式最小二乘法

根据如上图像进行分析可知当 $a_0 = 0, y = a_1x + a_2x^2$ 时拟合出来的更加合适。

4 习题四 指数函数最小二乘法

画图对比如下：

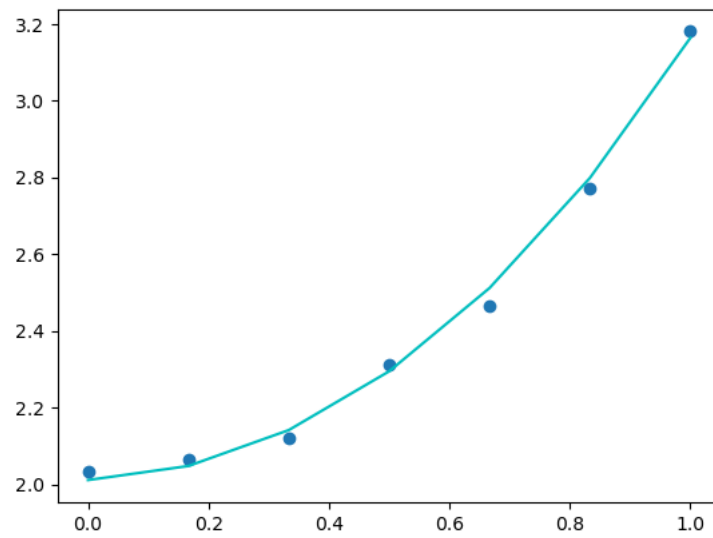


图 9: 指数函数最小二乘法