

实 验 报 告

实验地点		学生姓名	WJT
实验日期	2021 年 11 月 8 日 第 7、8 节	学院	数学与统计学院
实验课程	数值逼近	学号	
实验项目	三次样条插值	成绩	

一、实验目的或要求

1、编写三次样条插值程序

2、创新点

二、实验过程记录

以对函数 $y = e^x$ 进行插值为例，编写程序，插值点为区间 $[1,4]$ 均匀取 10 个点。待插值节点为区间 $[1,4]$ 均匀取 50 个点。

1、基础理论

step1: 根据函数给定的节点，计算 h_j ：

$$h_j = x_j - x_{j-1}$$

step2: 根据 h_j 计算 λ_j 、 μ_j ：

$$\lambda_j = \frac{h_{j+1}}{h_j + h_{j+1}}$$

$$\mu_j = 1 - \lambda_j$$

$$j = 1, \dots, N-1$$

step3: 添加边值条件

①第一类边值条件

$$s'(a) = dy_0, \quad s'(b) = dy_N$$

②第二类边值条件

$$s''(a) = d^2 y_0, \quad s''(b) = d^2 y_N$$

③自然边值条件

$$M_0 = 0, \quad M_N = 0$$

step4: 生成线性方程组，计算 M_j

step5: 对于 $x \in [x_{j-1}, x_j]$ ，代入公式

$$s(x) = M_{j-1} \frac{(x_j - x)^3}{6h_j} + M_j \frac{(x - x_{j-1})^3}{6h_j} + \left(y_{j-1} - \frac{M_{j-1}h_j^2}{6} \right) \frac{x_j - x}{h_j} + \left(y_j - \frac{M_j h_j^2}{6} \right) \frac{x - x_{j-1}}{h_j}$$

得到待插值点 x 对应的估计值。

2、代码

①计算 h_j

```
def H(X):
    ...
    计算h
    ...
    N = len(X)
    h = np.array([])
    for j in range(N-1):
        h = np.append(h, (X[j+1]-X[j]))
    return h
```

②计算 λ_j

```
def lamb(h):
    ...
    计算λ 1..N-1
    ...
    lam = np.array([])
    N = len(h)
    for j in range(N-1):
        j = j+1
        lam = np.append(lam, h[j]/(h[j-1]+h[j]))
    return lam
```

③计算 μ_j

```
def mu_(lam):
    ...
    计算mu 1..N-1
    ...
    mu = 1-lam
    return mu
```

④计算 d_j

```
def D(Y,h):
    ...
    计算d 1...N-1
    ...
    d = np.array([])
    N = len(h)
    for j in (np.arange(N-1)):
        j = j+1
        print(j)
        dd = 6*((Y[j+1]-Y[j])/h[j] - (Y[j]-Y[j-1])/h[j-1])/(h[j-1]+h[j])
        d = np.append(d, dd)
    return d
```

⑤添加边值条件（一类、二类、自然）

```

def side_cond(Y,h,lam,mu,d,cond='one',dy0=None,dyN=None):
    if cond=='one':
        ## 第一类边值条件
        lam0 = 1
        muN = 1
        d0 = 6/h[0]*((Y[1]-Y[0])/h[0]-dy0)
        dN = 6/h[-1]*(dyN-(Y[-1]-Y[-2])/h[-1])

        lam = np.insert(lam,0,lam0)
        mu = np.append(mu,muN)
        d = np.insert(d,0,d0)
        d = np.append(d,dN)
    elif cond=='natural':
        ## 自然边界条件
        lam0 = 0
        muN = 0
        d0 = 0
        dN = 0
        lam = np.insert(lam,0,lam0)
        mu = np.append(mu,muN)
        d = np.insert(d,0,d0)
        d = np.append(d,dN)
    elif cond=='two':
        ## 第二类边界条件
        lam0 = 0
        muN = 0
        d0 = 2*dy0
        dN = 2*dyN
        lam = np.insert(lam,0,lam0)
        mu = np.append(mu,muN)
        d = np.insert(d,0,d0)
        d = np.append(d,dN)
    return h,lam,mu,d

```

⑥生成线性方程组矩阵

```

def gen_matrices(lam,mu):
    n = len(lam)
    A = np.eye(n+1)*2
    for i in range(n):
        A[i,i+1] = lam[i]
        A[i+1,i] = mu[i]
    return A

```

⑦计算 M_j

```

def gen_M(A,d):
    d = d.reshape(-1,1)
    M = np.linalg.inv(A)@d
    return M

```

⑧三次样条插值函数

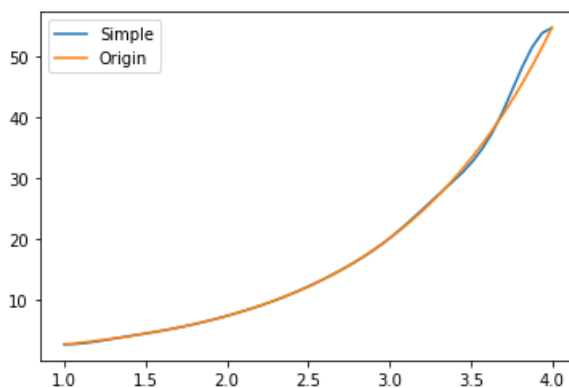
```
def three_simple(xs,X,Y,cond='natural',dy0=None,dyN=None):
    low = np.argsort(X)
    X = X[low]
    Y = Y[low]
    h = H(X)
    lam = lamb(h)
    mu = mu_(lam)
    d = D(Y,h)
    h,lam,mu,d = side_cond(Y,h,lam,mu,d,cond,dy0,dyN)
    A = gen_matrices(lam,mu)
    M = gen_M(A,d)
    S = np.array([])
    for x in xs:
        for i,Xi in enumerate(X):
            if x<Xi:
                j = i
                break
        s = M[j-1]*(X[j]-x)**3/(6*h[j-1])+M[j]*(x-X[j-1])**3/(6*h[j-1])+\\
            (Y[j-1]-M[j-1]*h[j-1]**2/6)*(X[j]-x)/h[j-1]+(Y[j]-M[j]*h[j]**2/6)*(x-X[j-1])/h[j-1]
        S = np.append(S,s)
    return S
```

输入待插值节点,插值节点及节点函数值,边界条件种类即可输出待插值节点的函数值。
三、实验结果报告及总结

1、利用第一类边值条件进行插值得到（默认 $dy_0=0,dy_N=0$ ）：

```
x = np.linspace(1,4,50)
y = three_simple(x,X,Y,cond='one',dy0=0,dyN=0)
```

```
plt.plot(x,y,label='Simple')
plt.plot(x,np.exp(x),label='Origin')
plt.legend(loc='best')
plt.show()
```

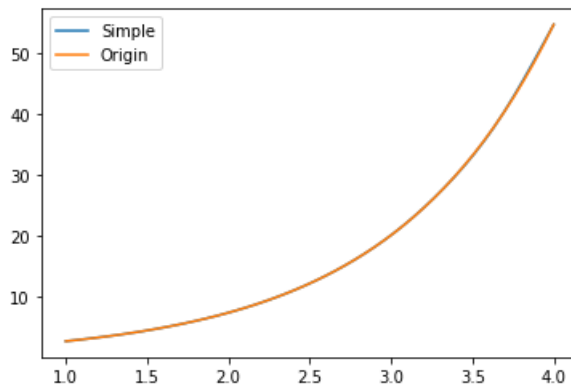


其中蓝线为插值函数，黄线为原函数。

2、利用第二类边值条件进行插值得到（默认 $d^2y_0=0,d^2y_N=0$ ）

```
x = np.linspace(1,4,50)
y = three_simple(x,X,Y,cond='two',dy0=0,dyN=0)
```

```
plt.plot(x,y,label='Simple')
plt.plot(x,np.exp(x),label='Origin')
plt.legend(loc='best')
plt.show()
```

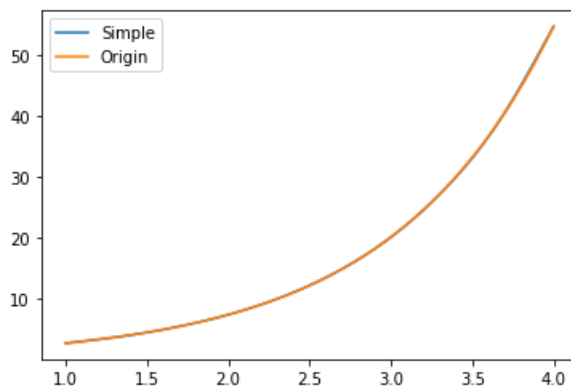


其中蓝线为插值函数，黄线为原函数。

3、利用自然边值条件得到

```
x = np.linspace(1,4,50)
y = three_simple(x,X,Y,cond='natural',dy0=0,dyN=0)
```

```
plt.plot(x,y,label='Simple')
plt.plot(x,np.exp(x),label='Origin')
plt.legend(loc='best')
plt.show()
```



其中蓝线为插值函数，黄线为原函数。

实验结果反思及讨论：
教师对报告的最终评价和意见： <div>年 月 日</div>