数值分析计算实习题实验报告

李竞宜　20214800

实现工具：Matlab

1. （第二章第1题）已知函数在下列各点的值为

x_i	0.2	0.4	0.6	0.8	1.0
$f(x_i)$	0.98	0.92	0.81	0.64	0.38

试用4次牛顿插值多项式$P_4(x)$及三次样条函数$S(x)$(自然边界条件)对数据进行插值，用图给出$\{(x_i, y_i), x_i = 0.2 + 0.08i, i = 0, 1, 11, 10\}$，$P_4(x)$及$S(x)$.

由于本题中给出的差值节点为等距节点，因此首先想到使用差分形式牛顿插值多项式进行插值计算，即

```
% 插值节点
h = 0.2;
X = 0.2 : h : 1.0;
Y = [0.98, 0.92, 0.81, 0.64, 0.38];

%% P4
% 计算各阶差分
n = length(X) - 1;
p = zeros(n+1,n+1);
p(:,1) = Y(:);
for k = 1 : n
    p(1:n-k+1,k+1) = diff(p(1:n-k+2,k));
end
q = p(1,:);
```

可以得到差分表，即

```
p =

    0.9800   -0.0600   -0.0500   -0.0100   -0.0200
    0.9200   -0.1100   -0.0600   -0.0300         0
    0.8100   -0.1700   -0.0900         0         0
    0.6400   -0.2600         0         0         0
    0.3800         0         0         0         0
```

进一步地，有

```
%计算p4(xi),差分形式
x = 0.2 * ones(1,4);
l = [0, 1, 11, 10];
x = x + 0.08*l;
t = zeros(1,4);
for k=1:4
    t(k) = (x(k)-X(1))/h;
end
```

```
y = zeros(1,4);
y = y + q(1);
for j=1:4
    z = 1;
    for k = 1:4 % 这里是 4，不是 n(=5)
        z = z*(t(j)-k+1)/k;
        y(j) = y(j) + z*q(k+1);
    end
end
disp('4次Newton插值的计算结果为')
y
```

由此可以得出需要计算的节点对应的插值结果，即

```
y =

    0.9800    0.9622    0.2403    0.3800
```

但是，仔细观察发现本题要求与三次样条形式的插值进行横向对比，那么使用差分形式所得到的结果表达式并不是那么直观，因此可考虑使用差商形式的插值多项式，即

```
syms x df d;
df(1) = 1;
d(1) = Y(1);
c = Y(:);
n = length(X);
for j=2:n
    for i=n:-1:j
        c(i) = (c(i)-c(i-1))/(X(i)-X(i-j+1));
    end
end
for i=2:n
    df(i) = df(i-1)*(x-X(i-1));
    d(i) = c(i)*df(i);
end
disp('4次Newton插值多项式为')
P4 = vpa(collect(sum(d)),5)
```

```
4次Newton插值多项式为

P4 =

- 0.52083*x^4 + 0.83333*x^3 - 1.1042*x^2 + 0.19167*x + 0.98
```

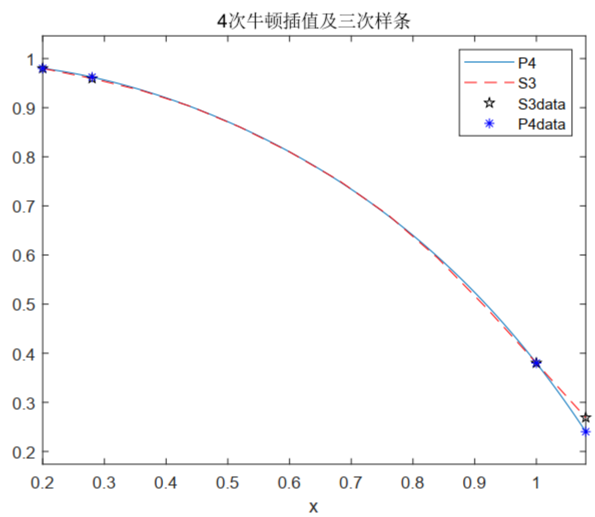
即4次牛顿插值多项式为$P_4(x) = -0.52083x^4 + 0.83333x^3 - 1.1042x^2 + 0.19167x + 0.98$

求解三次样条插值函数可直接调用matlab内部函数快捷实现，

```
syms x
disp('三次样条插值函数为')
pp = csape(X,Y, 'variational');%调用三次样条函数
q = pp.coefs;
for i=1:4
    S = q(i,:)*[(x-X(i))^3;(x-X(i))^2;(x-X(i));1];
    S = vpa(collect(S),5)
end
```

```
三次样条插值函数为

S =

- 1.3393*x^3 + 0.80357*x^2 - 0.40714*x + 1.04


S =

0.44643*x^3 - 1.3393*x^2 + 0.45*x + 0.92571


S =

- 1.6964*x^3 + 2.5179*x^2 - 1.8643*x + 1.3886


S =

2.5893*x^3 - 7.7679*x^2 + 6.3643*x - 0.80571
```

即三次样条插值函数$S_3(x) = \begin{cases} 1.3393x^3 + 0.80357x^2 - 0.40714 * x + 1.04 & x \in [0.2, 0.4] \\ 0.44643x^3 - 1.3393x^2 + 0.45x + 0.92571 & x \in [0.4, 0.6] \\ -1.6964x^3 + 2.5179x^2 - 1.8643x + 1.3886 & x \in [0.6, 0.8] \\ 2.5893x^3 - 7.7679x^2 + 6.3643x - 0.80571 & x \in [0.8, 1.0] \end{cases}$，图形对比结果如下



2. （第二章第3题）下列数据点的插值

x	0	1	4	9	16	25	36	49	64
y	0	1	2	3	4	5	6	7	8

可以得到平方根函数的近似，在区间$[0,64]$上作图.

(1) 用这9个点作8次多项式插值$L_8(x)$.

(2) 用三次样条（第一边界条件）程序求$S(x)$.

从得到的结果上在$[0,64]$上，哪个插值更精确；在区间$[0,1]$，两种插值哪个更精确？

(1) 根据拉格朗日插值多项式求解步骤，在导入数据后先计算其基函数，然后根据基函数得出拉格朗日多项式，具体为，

```
clear;clc;

% 插值节点
X = [0 1 4 9 16 25 36 49 64];
Y = [0 1 2 3 4 5 6 7 8];

%% L8
m = length(X);
L = ones(m,m);
for k = 1 : m
    V = 1;
    for i = 1 : m
        if k ~= i
            V = conv(V,poly(X(i))) / (X(k) - X(i));
        end
    end
    L1(k, :) = V;
    l(k, :) = poly2sym(V);
end
% fprintf('基函数为：\n');
% for k=1:m
%     fprintf('q%d(x)=%s\n',k,l(k));
% end
L = Y * l;
disp('拉格朗日多项式为')
L = vpa(collect(L),5)
L = matlabFunction(L);
```

运行后输出：

```
拉格朗日多项式为

L =

- 3.2806e-10*x^8 + 6.7127e-8*x^7 - 5.4292e-6*x^6 + 0.00022297*x^5 -
0.0049807*x^4 + 0.060429*x^3 - 0.38141*x^2 + 1.3257*x
```

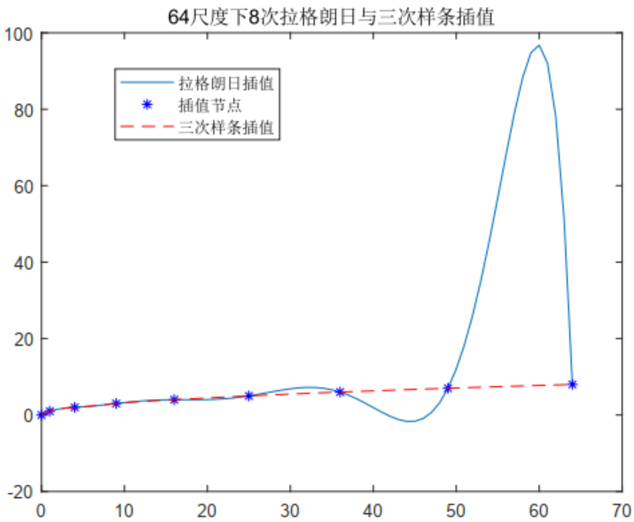
（2）三次样条插值与上题类似，这里不再赘述代码，直接给出插值结果

```
三次样条插值函数为

S =

- 0.086823*x^3 - 1.1102e-15*x^2 + 1.0868*x


S =

0.032042*x^3 - 0.35659*x^2 + 1.4434*x - 0.11886


S =

- 0.0027304*x^3 + 0.060678*x^2 - 0.22567*x + 2.1066


S =

0.00063484*x^3 - 0.030183*x^2 + 0.59208*x - 0.34667
```

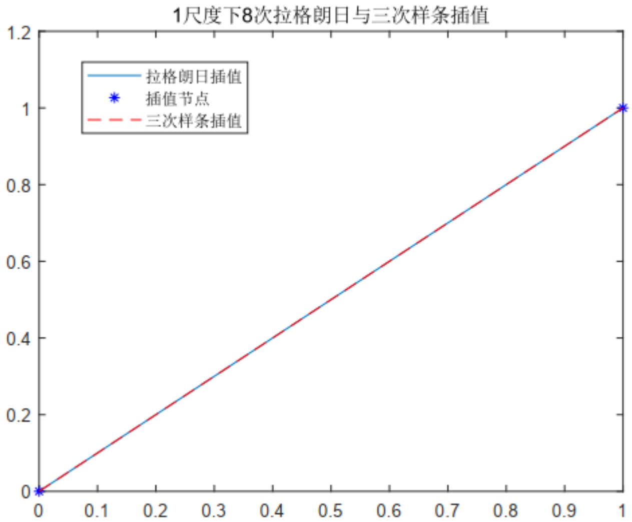
即三次样条插值多项式为

$$S_3(x)=\begin{cases}-0.086823x^3-1.1102e-15x^2+1.0868x & x\in[0,4]\\ 0.032042x^3-0.35659x^2+1.4434x-0.11886 & x\in[4,16]\\ -0.0027304x^3+0.060678x^2-0.22567x+2.1066 & x\in[16,36]\\ 0.00063484*x^3-0.030183*x^2+0.59208*x-0.34667 & x\in[36,49]\end{cases}$$

接着，我们观察在$[0,64]$尺度上，两种插值之间的对比，如下图所示



从图中可以看出，三次样条插值的插值效果相对较好，拉格朗日插值有较严重的龙格现象。进一步地，我们观察在$[0,1]$尺度上两种插值之间的对比，如下图所示



从视觉感官直接对比两者相差不大，不妨给出几组数值进行对比，取$x = [0.1 : 0.1 : 1]$,即

```
x4 = [0.1:0.1:1];
len_1 = length(x4);
y4 = zeros(1,len_1);
for i=1:len_1
    y4(i) = L(x4(i));
end
disp('准确值为')
sqrt(x4)
disp('拉格朗提插值结果为')
y4
disp('平均相对误差为')
e4 = abs((y4-sqrt(x4)))./sqrt(x4);
e4 = mean(e4(:))
disp('三次样条插值结果为')
y5 = fnval(pp,x4)
disp('平均相对误差为')
e5 = abs((y5-sqrt(x4)))./sqrt(x4);
e5 = mean(e5(:))
```

从以下结果中可以看出，从经验上，使用拉格朗日插值在$[0, 1]$区间内相对误差更小。

```
准确值为

ans =

    0.3162    0.4472    0.5477    0.6325    0.7071    0.7746    0.8367    0.8944
   0.9487    1.0000

拉格朗提插值结果为

y4 =

    0.1288    0.2504    0.3650    0.4730    0.5748    0.6706    0.7607    0.8455
   0.9251    1.0000

平均相对误差为

e4 =
```

```
    0.2110

三次样条插值结果为

y5 =

    0.1086    0.2167    0.3237    0.4292    0.5326    0.6333    0.7310    0.8250
   0.9148    1.0000

平均相对误差为

e5 =

    0.2571
```

3.（第三章第2题）由实验给出数据表

x	0.0	0.1	0.2	0.3	0.5	0.8	1.0
y	1.0	0.41	0.50	0.61	0.91	2.02	2.46

试求3次、4次多项式的曲线拟合，再根据数据曲线形状，求一个另外函数的拟合曲线，用图示数据曲线及相应的三种拟合曲线.

（1）求以上数据的3次、4次多项式拟合

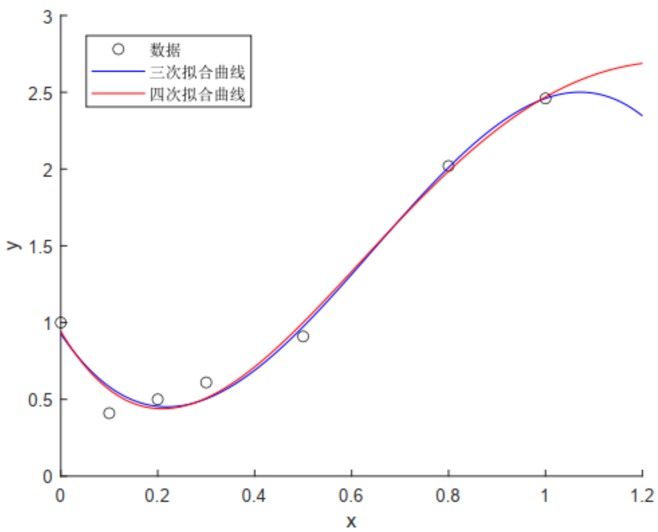
求多项式拟合可以直接调用Matlab中的多项式拟合函数，即

```
clc;clear;
%原始数据
x=[0,0.1,0.2,0.3,0.5,0.8,1.0];
y=[1.0,0.41,0.50,0.61,0.91,2.02,2.46];
scatter(x,y,'k');               %画出原始数据散点图
hold on

%调用内部函数拟合
fitting = polyfit(x,y,3); %3次拟合  降幂排列
disp('三次拟合降幂排列系数为')
fitting
xi = 0:0.001:1.2;
yi = polyval(fitting,xi);    %多项式求值
plot(xi,yi,'b');           %观测数据点
hold on

fitting2 = polyfit(x,y,4); %4次拟合  降幂排列
disp('四次拟合降幂排列系数为')
fitting2
xj = 0:0.001:1.2;
yj = polyval(fitting2,xj);    %多项式求值
plot(xj,yj,'r');

xlabel('x');
ylabel('y');
```

其输出的结果为

```
三次拟合降幂排列系数为

fitting =

   -6.6221    12.8147    -4.6591     0.9266

四次拟合降幂排列系数为

fitting2 =

    2.8853   -12.3348    16.2747    -5.2987     0.9427
```

即，三次拟合函数为$y = -6.6221 - 12.8147x - 4.6591x^2 - 0.9266x^3$，四次拟合函数为$y = 2.8853 + -12.3348x + 16.2747x^2 - 5.2987x^3 + 0.9427x^4$,讲两条曲线与原数据绘制在同一坐标轴下对比，如下图所示

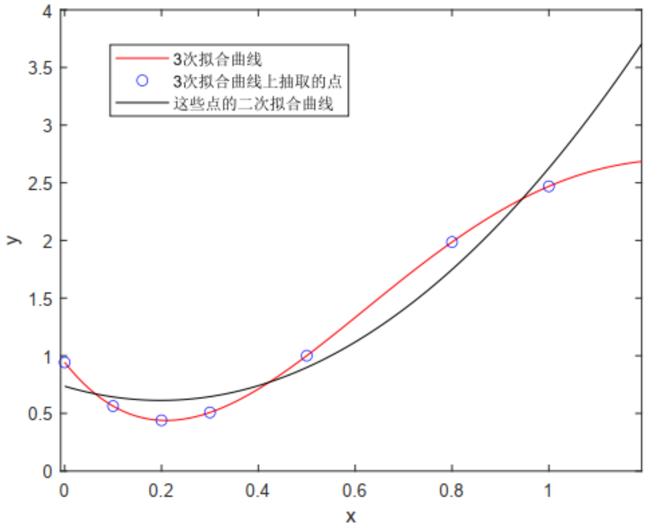


（2）求另一个拟合曲线

原题目中表意不是特别清晰，我根据我理解去"求另一个拟合曲线"，思路是，按照上面求得的4次拟合曲线，取曲线上的一些点，然后根据这些进行2次拟合观察，具体为

```
%% 求另外曲线拟合
xk = [0,0.1,0.2,0.3,0.5,0.8,1.0];
yk = zeros(1, length(xk));
for i = 1:length(xk)
    yk(i) = polyval(fitting2, xk(i));
end
scatter(xk,yk,'b');              %画出生成数据散点图
hold on

fitting3 = polyfit(xk,yk,2); %2次拟合  降幂排列
disp('抛物线拟合降幂排列系数为')
fitting3
xl = 0:0.001:1.2;
yl = polyval(fitting3,xl);    %多项式求值
plot(xl,yl,'k');
```

其结果如下图所示：



4. （第五章第3题）线性方程组$Ax=b$的A及b为

$$A=\begin{bmatrix}10&7&8&7\\7&5&6&5\\8&6&10&9\\7&5&9&10\end{bmatrix}\quad b=\begin{bmatrix}32\\23\\33\\31\end{bmatrix}$$

则解$x=(1,1,1,1)^T$. 用MATLAB内部函数求$det A$及A的所有特征值和$cond(A)_2$.若令

$$A+\delta A=\begin{bmatrix}10&7&8.1&7.2\\7.08&5.04&6&5\\8&5.98&9.89&9\\6.99&5&9&9.98\end{bmatrix}$$

求解$(A+\delta A)(x+\delta x)=b$，输出向量$\delta x$和$\|\delta x\|_2$. 从理论和实际计算两方面分析线性方程组$Ax=b$解的相对误差$\|\delta x\|_2/\|x\|_2$及$A$相对误差$\|\delta A\|_2/\|A\|_2$.

（1）求$det A$及A的所有特征值和$cond(A)_2$

```
clc;clear
A = [10 7 8 7;7 5 6 5;8 6 10 9;7 5 9 10];
b = [32 23 33 31]';
x0 = [1 1 1 1]';
%% 求解detA、cond（A）、A的特征值
det(A)
cond(A,2)
eig(A)
```

输出结果为

```
ans =

    1.0000


ans =

   2.9841e+03


ans =
```

```
    0.0102
    0.8431
    3.8581
   30.2887
```

（2）扰动分析

```
%% 干扰分析
newA = [10 7 8.1 7.2;7.08 5.04 6 5;8 5.98 9.89 9;6.99 5 9 9.98];
x = newA \ b; %扰动下的解
del_x = x - x0; %扰动对解的影响
norm(del_x,2);
del_A = newA-A

eA = norm(del_A,2)/norm(A,2)
ex = norm(del_x,2)/norm(x0,2)
```

输出结果为

```
eA =

    0.0076


ex =

   10.4661
```

从输出实际计算结果可见，A的相对误差仅仅只有0.0076的情况下，结果x的相对误差就达到了10.4661，说明该方程组是病态的；A的条件数2954.1远远大于1，根据条件数判断方程组病态与否的理论，在如此条件数下，当A只有很小的误差就会给结果带来很大影响。

Remark:

根据本报告指引应该可以完成所有结果复现，而完整代码可访问本人github：https://github.com/Drenches/Numerical-Analysis-Project.