2.1 最佳平方逼近多项式

一般情况下，记已知函数 $f(x)$ 关于权函数 $\rho(x)$ 的最佳平方逼近多项式为

$p(x)=a_0+a_1x+\cdots+a_nx^n$，由最佳平方逼近的定义有：

$$\frac{\partial I}{\partial a_k}=0 \qquad k=0,1,\cdots,n$$

其中 $I(a_0,a_1,\cdots,a_n)=\int_a^b \rho(x)[f(x)-\sum_{j=0}^{n}a_jx^j]^2dx$ 。整理为求解 $p(x)$ 的系数

$a=[a_0,a_1,\cdots,a_n]^T$ 的方程组 $Ca=d$，其中

$$C=\begin{bmatrix}\mu_0 & \mu_1 & \cdots & \mu_n\\ \mu_1 & \mu_2 & \cdots & \mu_{n+1}\\ \vdots & \vdots & \ddots & \vdots\\ \mu_n & \mu_{n+1} & \cdots & \mu_{2n}\end{bmatrix} \qquad d=\begin{bmatrix}d_0\\ d_1\\ \vdots\\ d_n\end{bmatrix}, \qquad \begin{aligned}\mu_i&=\int_a^b \rho(x)x^i dx\\ d_i&=\int_a^b \rho(x)f(x)x^i dx\end{aligned}$$

利用某些特殊的正交多项式，可以比较方便的实现函数在固定区间上关于特殊权函数的最佳平方逼近多项式。如Legendre多项式 $\{P_0(x),P_1(x),\cdots,P_n(x)\}$ 为 $[-1,1]$ 上关于权函数 $\rho(x)=1$ 的正交多项式，则函数 $f(x)\in C[-1,1]$ 于权函数 $\rho(x)=1$ 的最佳平方逼近多项式

为 $$S_n(x)=a_0P_0(x)+a_1P_1(x)+\cdots+a_nP_n(x)$$

其中 $$a_k=\frac{2k+1}{2}\int_{-1}^{1}f(x)P_k(x)dx$$

Matlab 代码如下：

```
function [f,f0] = Legendre(func,n,x0)
%求函数在[-1,1]的关于权函数为1的n次最佳平方逼近多项式f，并计算插值多项式f在数据点x0的函数值f0
syms t;
P(1:n+1) = t;
P(1) = 1;
P(2) = t;
c(1:n+1) = 0.0;
c(1)=int(subs(func,findsym(sym(func)),sym('t'))*P(1),t,-1,1)/2;
c(2)=3*int(subs(func,findsym(sym(func)),sym('t'))*P(2),t,-1,1)/2;
f = c(1)+c(2)*t;
for i=3:n+1
```

```
    P(i) = ((2*i-3)*P(i-1)*t-(i-2)*P(i-2))/(i-1);
    c(i) = (2*i-1)*int(subs(func,findsym(sym(func)),t)*P(i),t,-1,1)/2;
    f = f + c(i)*P(i);
   if(i==n+1)
      f0 = subs(f,'t',x0);
      f = vpa(f,6);
    end
end
```

2.2 最小二乘法

若函数 $f(x) \in C[a,b]$ 只在一组离散点集 $\{x_i, i=0,1,\cdots,m\}$ 上给出，即常见的实验数据 $\{(x_i,y_i), i=0,1,\cdots,m\}$ 的多项式曲线拟合。一般的，在最小二乘法中，使多项式

$p(x)=\sum_{j=0}^{n} a_j x^j$ 满足

$$I(a_0,a_1,\cdots,a_n)=\sum_{i=0}^{m}\omega(x_i)[\sum_{j=0}^{n}a_j x_i^j - f(x_i)]^2$$

达到最小，即 $\dfrac{\partial I}{\partial a_k}=0\ (k=0,1,\cdots,n)$。整理为求解 $p(x)$ 的系数 $a=[a_0,a_1,\cdots,a_n]^T$ 的方程组 $Ga=d$，其中

$$G=\begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_n & \mu_{n+1} & \cdots & \mu_{2n} \end{bmatrix} \qquad d=\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix}, \qquad \begin{aligned} \mu_k &= \sum_{i=0}^{m}\omega(x_i)x_i^k \\ d_k &= \sum_{i=0}^{m}\omega(x_i)f(x_i)x_i^k \end{aligned}$$

Matlab 代码如下：

```
function f=multifit(x,y,wfunc,n)
%x,y为给定数据的数组，wfunc为权函数，n为拟合多项式的次数
N=length(x);
M=length(y);
if(N ~= M)
    disp('x与y维数不匹配');
    return;
end
var = findsym(sym(wfunc));
w = subs(wfunc,’var’,x);
g(1:(2*n+1))=0;
b(1:(n+1))=0;
```

```
for j=1:(2*n+1)
    for k=1:N
        g(j)=g(j)+w(j)*x(k)^(j-1);
        if(j<(n+2))
            b(j)=b(j)+w(j)*y(k)*x(k)^(j-1);
        end
    end
end

G(1,:)=g(1:(n+1));
for i=2:(n+1)
    G(i,:)=g(i:(n+i));
end
coff=b'\G;
f = coff(1);
l = 1;
for i=1:n
    l = l*t;
    f = f+coff(i+1)*l;
end
```