浙江大学

数 值 分 析 方 法

题目：函数拟合的 GUI 实现

姓名学号：szx 3210000000

指导老师：余光定

年级及专业：大二　电子科学与技术

学院：信电学院

一、设计思路

1. GUI 设计界面搭建

借用 Matlab 的 App Designer 设计工具，使用内置的 Table、Button、输入框、二维坐标图搭建 GUI 图形界面，在 Button 上设置回调函数，与用户进行交互，输入框用于给用户输入拟合相关参数。

2. 数据输入及导入

程序使用 Excel 文件输入数据。用户通过在指定的 Excel 文件中输入待拟合的数据点，由程序中的 Read 按钮对同目录下的 Excel 文件数据进行读取，并在对数据进行排序后显示在 Table 中。输入的数据保存在全局变量中，方便拟合函数的调用。

3. 函数拟合

程序提供三种函数拟合方式：拉格朗日内插、三次样条和多项式拟合。三种拟合方式分立在不同页面中，数据由同一 Excel 文件提供，但回调、数据调用、结果显示等操作独立进行。在拟合算法函数中，通过对数据的运算，得到拟合的多项式的系数及符号表达式，并输出到回调函数中。

4. 表达式输出及图像绘制

用户在点击 Button 后，回调函数获得拟合结果的符号表达式，使用 fplot 函数，使函数在指定的坐标图中绘制，并对函数图像的绘制区间进行限定。将函数符号表达式转换位向量式，并在数字编辑框中显示输出。

二、算法描述

1. 拉格朗日内插法

算法基本思想：

构造 n 次多项式，能够经过所有的数据点，充分利用数据点，并使 n 的次数较少，使得拟合的函数连续，且包含所给定的所有信息。

算法的数学表达：

$$\mathrm{L}_{n,k}(x)=\prod_{i=0,i\neq k}^{n}\frac{x-x_i}{x_k-x_i}$$

该式子对所有的数据都纳入其中，对给定的 n+1 个点构造 n 次表达式，并使得拟合函数满足：

$$L_{n,k}(x_i)=\left\{\begin{matrix}1, i=k\\0, otherwise\end{matrix}\right\}$$

最后，对每一数据点进行相同操作并相加，得到：

$$F(x)=\sum_{k=0}^{n}f(x_k)L_{n,k}(x)$$

代码实现：

```
function funL=Lagrange(~, xx, yy) %拉格朗日内插法的算法代码
    syms x;
    n = length(xx);
    f=0;
    L=1;
    for i=1:n
        for j=1:n
            if j~=i
                L=L*(x-xx(j))/(xx(i)-xx(j));
            end
        end
        f=f+L*yy(i);
        L=1;
    end
    funL=simplify(f);
end
```

定义符号 x，n 为数据点个数，对每一数据进行求 L 的操作（两层 for 循环运算），最后乘以 y 值并相加，得到 F 拟合函数，输出到回调函数。funL 即为化简后的拉格朗日内插函数表达式，xx、yy 分别为输入的数据的 x、y 值。

2. 三次样条拟合

算法基本思想：

对给定数据，每两点之间构造三次多项式，保证多项式端点值都与对应的数据点值相同，即保证分段函数连续，并使端点处一阶导数、二阶导数相等。对最外围两点，在无要求时，二阶导数都为 0；在有要求即设为给定值。再次基础上，对每一段函数，都有四个限制条件，可解出所有系数。

算法的数学表达：

每段函数设为：

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

令 $h_j = x_{j+1} - x_j$，每段满足的方程为：

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$$

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2$$

$$c_{j+1} = c_j + 3d_j h_j$$

对上式进行化简和组合，可得到矩阵方程：

$$AX = B$$

$$A = \begin{bmatrix} 1 & 0 & 0 & & & 0 \\ \mathrm{h}_0 & 2(h_0 + h_1) & h_1 & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \mathrm{h}_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & & & 0 & 0 & 1 \end{bmatrix}$$

$$\mathrm{B} = \begin{bmatrix} 0 \\ \dfrac{3}{h_1}(a_2 - a_1) - \dfrac{3}{h_0}(a_1 - a_0) \\ \cdots \\ \dfrac{3}{h_{n-1}}(a_n - a_{n-1}) - \dfrac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix}$$

$$X=\begin{bmatrix} c_0 \\ c_1 \\ \cdots \\ c_n \end{bmatrix}$$

由此，可计算得到拟合多项式的系数的矩阵形式。

当对端点二阶导数有要求时，

$$A=\begin{bmatrix} 2\mathrm{h}_0 & \mathrm{h}_0 & 0 & & & 0 \\ \mathrm{h}_0 & 2(h_0+h_1) & h_1 & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \mathrm{h}_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & & & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix}$$

$$\mathrm{B}=\begin{bmatrix} \frac{3}{h_0}(a_1-a_0)-3f'(a) \\ \frac{3}{h_1}(a_2-a_1)-\frac{3}{h_0}(a_1-a_0) \\ \cdots \\ \frac{3}{h_{n-1}}(a_n-a_{n-1})-\frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\ 3f'(b)-\frac{3}{h_{n-1}}(a_n-a_{n-1}) \end{bmatrix}$$

代码实现：

```
function TCSI = Cubic1(~, x, y) %三次样条（自然）的算法代码，得到系数矩阵
    n=length(x);
    a=y;
    b=zeros(n-1,1);
    d=zeros(n-1, 1);
    h=diff(x);
    B=zeros(n, 1);
    A=zeros(n, n);
    B(1)=0;
    B(n)=0;

    for i=2:n-1
        B(i)=(3/h(i))*(a(i+1)-a(i))-(3/h(i-1))*(a(i)-a(i-1));
    end

    A(1, 1)=1;
    A(n, n)=1;
    for i=2:n-1
        A(i,i-1)=h(i-1);
        A(i,i)=2*(h(i-1)+h(i));
        A(i,i+1)=h(i);
    end

    c=A\B;
    for i=1:n-1
        b(i)=(a(i+1)-a(i))/h(i)-h(i)*(c(i+1)+2*c(i))/3;
        d(i)=(c(i+1)-c(i))/(3*h(i));
    end

    aa=zeros(n-1, 1);
    cc=zeros(n-1, 1);
    for i=1:n-1
        aa(i)=a(i);
        cc(i)=c(i);
    end

    TCSI=[d cc b aa];
end
```

```
function TCSI = Cubic2(~, x, y, L, R) %三系样条（限制）的算法，得到系数矩阵
    n=length(x);
    a=y;
    b=zeros(n-1,1);
    d=zeros(n-1, 1);
    h=diff(x);
    B=zeros(n, 1);
    A=zeros(n, n);
    B(1)=3*(a(2)-a(1))/h(1)-3*L;
    B(n)=3*R-3*(a(n)-a(n-1))/h(n-1);

    for i=2:n-1
        B(i)=(3/h(i))*(a(i+1)-a(i))-(3/h(i-1))*(a(i)-a(i-1));
    end

    A(1, 1)=2*h(1);
    A(1, 2)=h(1);
    A(n, n-1)=h(n-1);
    A(n, n)=2*h(n-1);
    for i=2:n-1
        A(i,i-1)=h(i-1);
        A(i,i)=2*(h(i-1)+h(i));
        A(i,i+1)=h(i);
    end

    c=A\B;
    for i=1:n-1
        b(i)=(a(i+1)-a(i))/h(i)-h(i)*(c(i+1)+2*c(i))/3;
        d(i)=(c(i+1)-c(i))/(3*h(i));
    end

    aa=zeros(n-1, 1);
    cc=zeros(n-1, 1);
    for i=1:n-1
        aa(i)=a(i);
        cc(i)=c(i);
    end
```

先定义出系数矩阵，通过循环写出运算矩阵，进行计算后，将系数矩阵返回，并赋值给 Table。

3. 多项式近似

算法基本思想：

对于给定的目标多项式阶数，将 x 值代入拟合表达式，并于对应的函数值比较，选择合适的系数，使得其方差最小，从而使得在给定阶数下，函数拟合的偏差最小。

算法的数学表达：

$$E=\sum_{i=1}^{m}(y_i-P_n(x_i))^2$$

该式为方差的表达式，P 为拟合函数，我们要使得 E 最小，故对其进行求导运算得：

$$0=\frac{\partial E}{\partial a_j}=-2\sum_{i=1}^{m}y_i x_i^j+2\sum_{k=0}^{n}a_k\sum_{i=1}^{m}x_i^{j+k}$$

由此，移项化简得到运算式：

$$\sum_{k=0}^{n}a_k\sum_{i=1}^{m}x_i^{j+k}=\sum_{i=1}^{m}y_i x_i^j$$

从中，我们便可算出拟合函数的系数。

代码实现：

```
function funDLS = Squares(~, xx, yy, n) %多项式近似的算法代码
    m=length(xx);
    if n>m
        funDLS=0;
    else
        sum=0;
        a=zeros(n+1, n+1);
        b=zeros(n+1,1);
        for i=1:n+1
            for j=1:n+1
                for k=1:m
                    sum=sum+power(xx(k),i+j-2);
                end
                a(i, j)=sum;
                sum=0;
            end
        end
        sum=0;
        for i=1:n+1
            for k=1:n
                sum=sum+power(xx(k),i-1)*yy(k);
            end
            b(i)=sum;
            sum=0;
        end
        c=a\b;
        syms x;
        y=0;
        for i=1:n+1
            y=y+c(i)*power(x, i-1);
        end
        funDLS=vpa(y,4);
    end
end
```

代码思路与算法相同，通过循环得到运算表达式，计算得到系数，再通过循环相加，得到符号表达式。

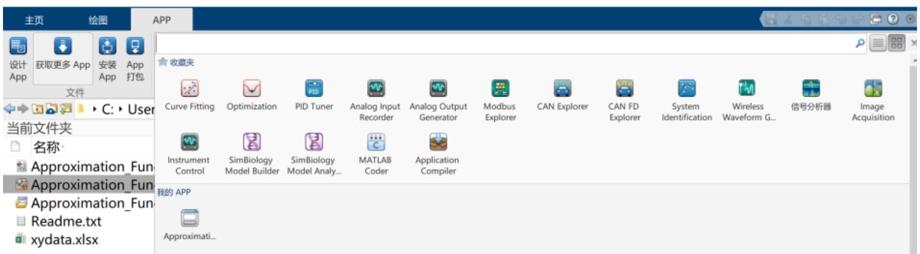
三、软件用法

1. Approximation_Function.mlapp 文件

打开 matlab，在该文件所在目录下，打开该文件，点击运行即可使用。

2. Approximation_Function.mlappinstall 文件

打开 matlab，点击菜单中的 APP 按钮，再点击 APP 安装，选择该文件，即可将该 APP 安装进“我的 APP”中。

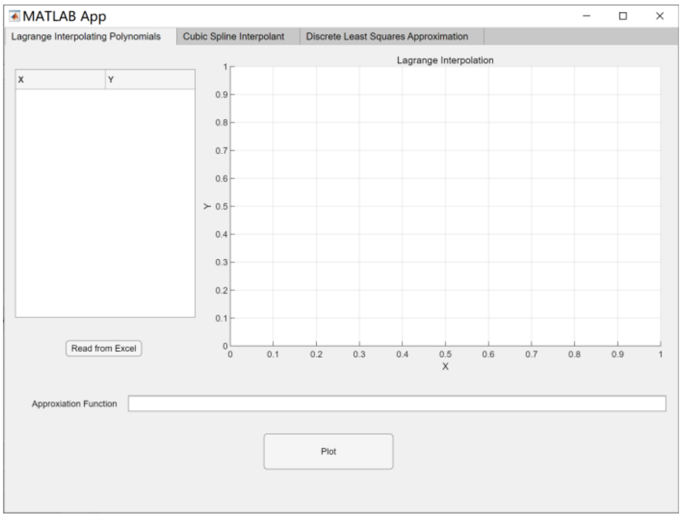


3. 数据输入及修改

应用程序通过 xydata.xlsx 文件读入数据，输入或修改数据直接在该文件中相应栏输入或修改数据即可，支持乱序输入。

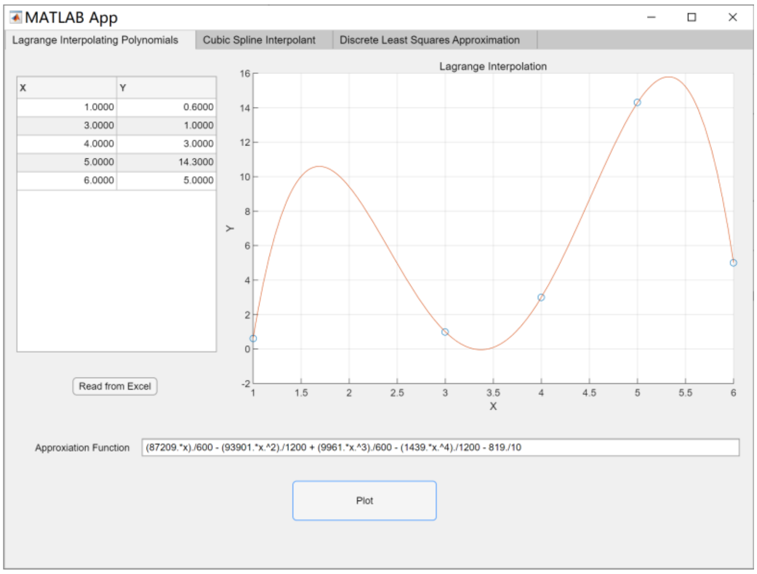
由于程序读取文件的路径已在程序代码中固定，所以请不要移动 Excel 文件的位置，以免程序不能正常读取。

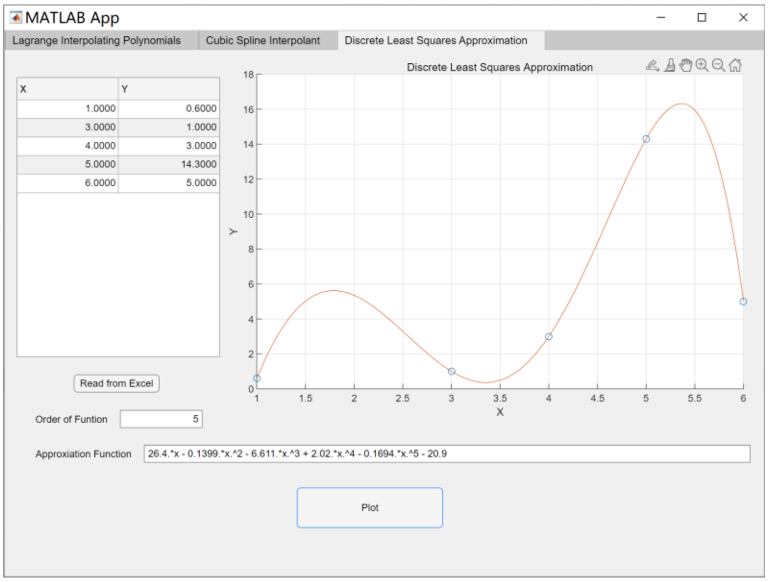
4. 应用程序使用

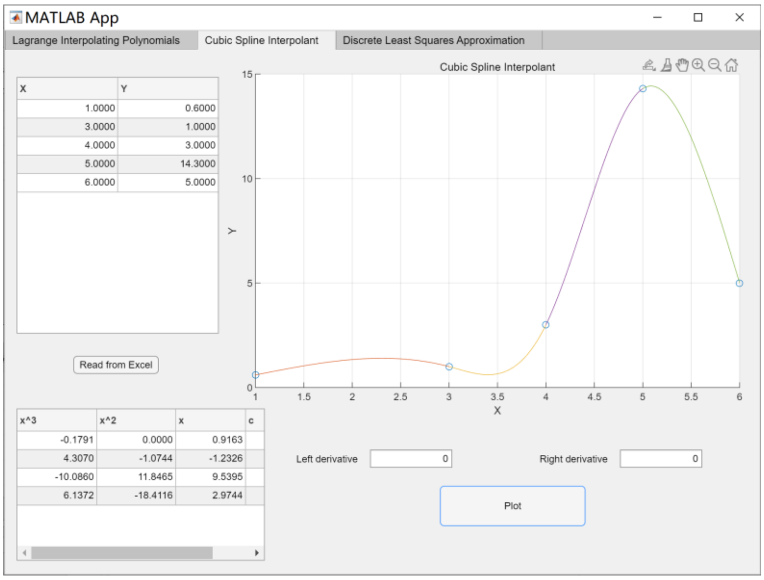


三种拟合方法分立于三个独立的界面，通过程序上方的按钮进行切换。

在拟合函数时，先点击“Read from Excel”按钮，再点击“Plot”按钮。函数图像会绘制于坐标图内，拉格朗日内插和多项式拟合的结果会在输出框中显示；三次样条的拟合函数的系数会列于 Table 中。

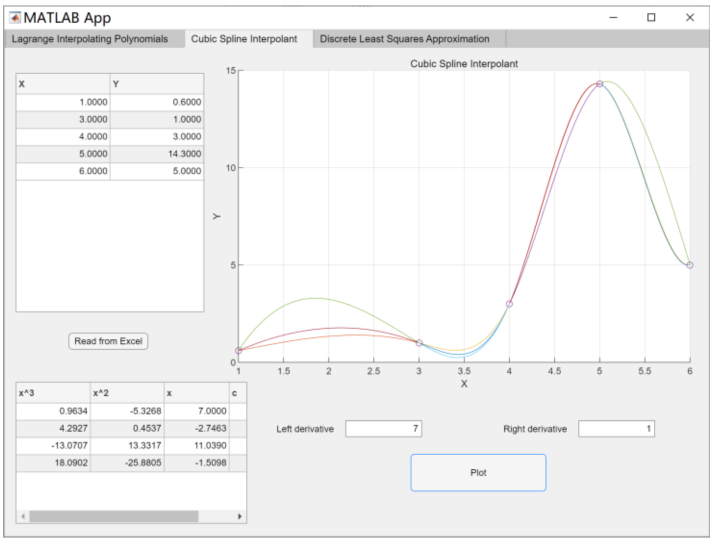






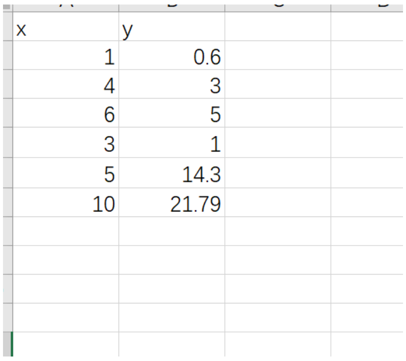
三次样条的端点限制和多项式的阶数输入都通过数字输入框输入。

程序支持不同限制条件下，拟合函数图像的对比。



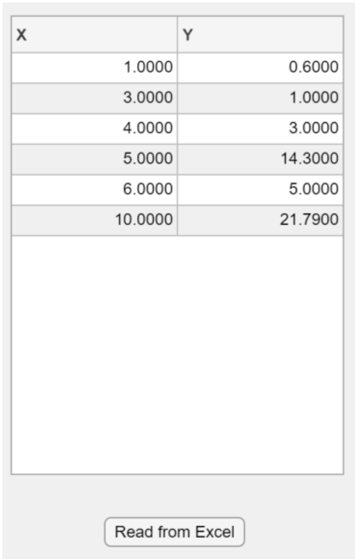
四、应用示例

使用的数据已经置于 xydata.xlsx 文件中



x	y
1	0.6
4	3
6	5
3	1
5	14.3
10	21.79

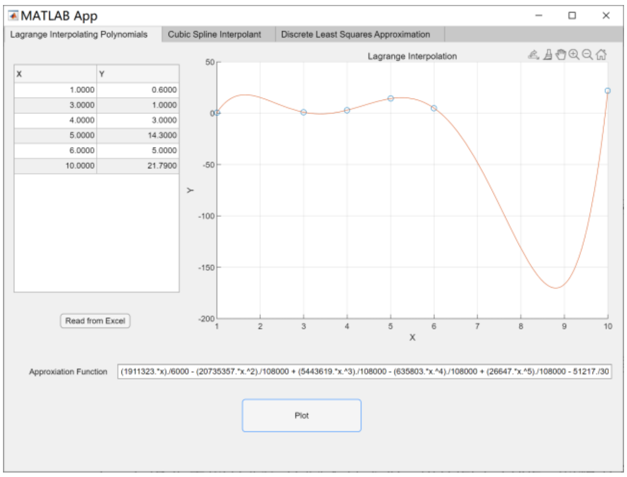
运行程序后，点击“Read from Excel”按钮，数据被排序，并显示在 Table 中。



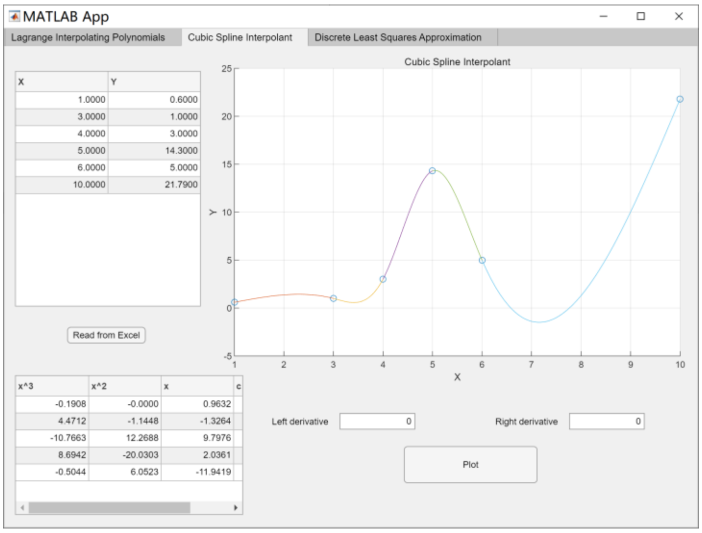
X	Y
1.0000	0.6000
3.0000	1.0000
4.0000	3.0000
5.0000	14.3000
6.0000	5.0000
10.0000	21.7900

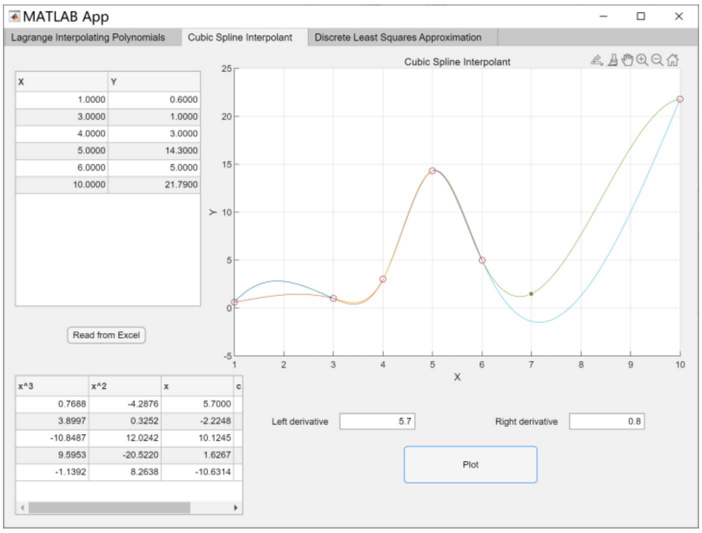
之后，再点击“Plot”按钮，就可得到函数图像和拟合表达式。

拉格朗日内插：

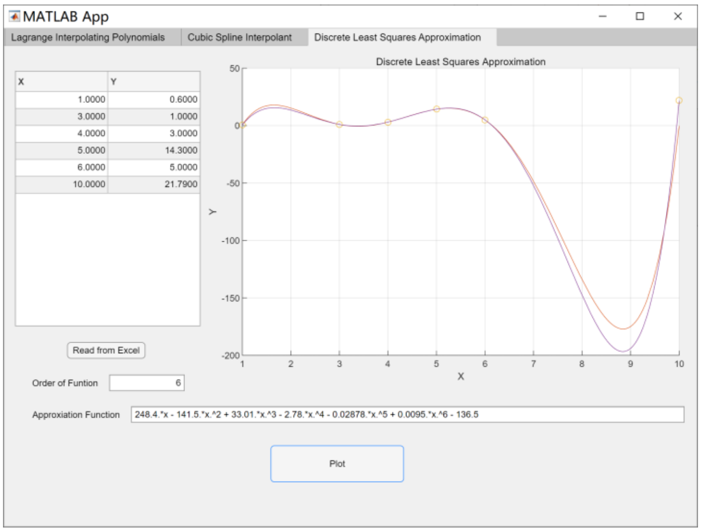


三次样条（自然与有限制）：





多项式近似（五次及六次）：



图形及表达式都成功显示，程序基本是成功的

五、性能分析

1. 计算复杂度

拉格朗日内插：使用了两层 for 循环，复杂度应为 n^2。

三次样条：使用一层 for 循环及矩阵运算，复杂度应为 n。

多项式近似：使用三层 for 循环，复杂度应为 n^3。

2. 运行速度

拉格朗日内插：0.62s

三次样条：0.87s

多项式近似：0.76s

3. 准确性

拟合函数的准确性由算法决定，三种算法各有利弊。

拉格朗日内插：

误差为：

$$\frac{f^{(n+1)}(\delta(x))}{(n+1)!}\prod_{i=0}^{n} x - x_i$$

并且拟合出的函数次数较高，在数据点本身有误差时，没办法排除。但胜在算法简单，充分利用数据，表达简洁。

三次样条：

拟合的精确度比较高，对数据充分地利用，拟合函数次数较低。但是，运算比较复杂，并且拟合出的函数并不是完全连续的，只在二阶内连续，对于高阶的问题无法解决。

多项式拟合：

阶数给定，可以得到具有特定关系的变量之间的函数系数，并且可以排除数据点中误差较大的点。缺点在于通常不过数据点，在大区域内的关系能够很好地表达，但对于小范围的拟合精度较低。