学号 WA2214014 专业 人工智能 姓名 杨跃浙
实验日期 25.01.03 教师签字 成绩

实验报告

【实验名称】 实验三

【实验目的】

一、插值方法掌握与应用

1. 熟练掌握 Newton 插值法和 Lagrange 插值法的原理及计算流程，能够根据给定的数据表准确构建差商表，并运用一次和二次 Newton 插值多项式以及 Lagrange 插值法对指定点进行函数近似值计算，深入理解不同插值方法在数据处理中的特点和适用场景，提升数值计算能力与数据逼近技巧。

2. 通过对 Lagrange 插值法在不同节点数量下的应用及对 Runge 现象的研究，探究插值多项式阶数和节点分布对插值结果准确性的影响机制，明确数值计算中可能出现的问题及应对策略，增强对插值误差的分析和控制能力，为实际工程和科学

计算中的数据插值应用提供理论与实践基础。

二、曲线拟合技能提升

1. 学会运用最小二乘法求解二次拟合曲线和特定形式拟合曲线，掌握根据给定数据构建法方程组并求解拟合系数的方法，能够准确得到拟合曲线方程，理解曲线拟合在数据建模和趋势分析中的重要作用，培养利用数学模型对实际数据进行有效描述和预测的能力。

2. 通过绘制原始数据点和拟合曲线，直观对比拟合效果，掌握评估拟合质量的方法，能够根据数据特点和实际需求选择合适的拟合函数形式，提高数据处理和分析的科学性与准确性，为后续的数据分析和挖掘工作奠定基础。

三、样条插值原理实践

深入理解三次样条插值函数的概念和求解方法，掌握在给定边界条件下利用工具函数或算法求解样条插值函数的过程，能够绘制样条插值曲线并分析其在区间内的光滑性和对原始数据的逼近性能，熟悉样条插值在处理复杂数据和保证曲线连续性方面的优势，拓展数值计算方法的应用领域和实践技能。

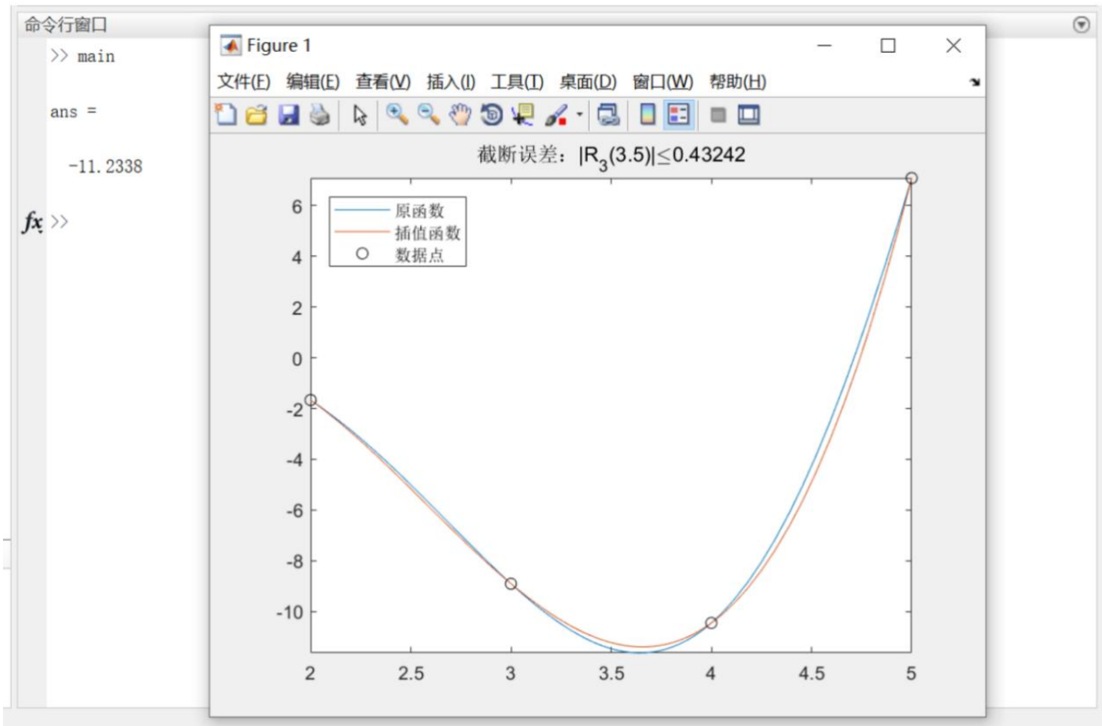
四、综合素养培养

通过对各类数值计算算法及案例的学习和实践，全面提升对数值分析领域的认知水平，培养独立思考、问题解决和编程实现的综合能力，能够将所学知识灵活应用于不同的实际问题，积累数值计算实践经验，形成严谨的科学思维和良好的实验习惯，为深入学习和研究相关领域奠定坚实的基础。

例：利用函数$f(x)=x^2\cos x$生成区间[2,5]上的4个等距离散点$(x_i, f(x_i))$, i=1,2,3，试利用Lagrange插值法计算$f(3.5)$的值并估计其截断误差.

```
f=@(x)x.^2.*cos(x);  % 原函数
n=4;  % 等分数据点数
xdata=linspace(2,5,n);  % x轴等分数据点
ydata=f(xdata);  % 等分数据点处的函数值
y=lagrange_interp(xdata,ydata);  % Lagrange插值
fplot(@(x)[f(x),polyval(y,x)],[2,5])  % 绘制原函数及插值多项式曲线
hold on  % 图形保持
plot(xdata,ydata,'ko')  % 绘制数据点
legend({'原函数','插值函数','数据点'},'location','northwest')  % 添加图例
dnf=diff(sym(f),n);  % 求原函数的n阶导数
dnfun=matlabFunction(dnf);  % 将符号函数转换为匿名函数
x0=fminbnd(@(x)-dnfun(x),2,5);  % 计算n阶导函数的极大值点
M=dnfun(x0);  % 计算4阶导数的极大值
xi=3.5;  % 插值数据点
polyval(y,xi)  % 计算插值数据点函数值
R=abs(M/gamma(n+1)*prod(xi-xdata));  % 截断误差的最大值
title(['截断误差: |R_',int2str(n-1),'(',num2str(xi,'%g'),...
    ')|\leq',num2str(R)])  % 添加标题
```

```
function varargout=lagrange_interp(xdata,ydata,xi)
% LAGRANGE_INTERP   Lagrange插值
n=length(xdata);  % 求向量xdata的长度
if length(unique(xdata))<n  % 若输入的点出现相同时，给出错误提示
    error('输入的点必须是互异的.')
end
L=zeros(n);  % 存储插值基函数
for i=1:n
    px=poly(xdata([1:i-1,i+1:n]));  % 构造以x_j为根的多项式(j=1:i-1,i+1:n)
    L(i,:)=px/polyval(px,xdata(i));  % 求插值基函数并存储
end
y=sum(bsxfun(@times,L,ydata(:)));  % 求插值多项式
if nargin==3  % 若输入参数为3个
    y=polyval(y,xi);  % 根据插值多项式求指定点处的值
end
[varargout{1:2}]=deal(y,...  % 第一个输出参数为插值多项式或其在某点处的值
    L);  % 第二个输出参数为插值基函数
```



例2.给定数表

x	1	2	4	5
y	3	6	2	1

x	y	$f[?,?]$	$f[?,?,?]$	$f[?,?,?,?]$
1	3	3	$-5/3$	$1/2$
2	6	-2	$1/3$	
4	2	-1		
5	1			

2.试用一次和二次Newton插值多项式计算*f(2.4)*的近似值。

2. 用一次Newton插值近似计算*f(2.4)*，应选与2.4最近的2个节点：

$$x_0 = 2, x_1 = 4$$

由表中 行数据有 $\boldsymbol{N_1(x) = 6 - 2(x-2) = -2x + 10}$

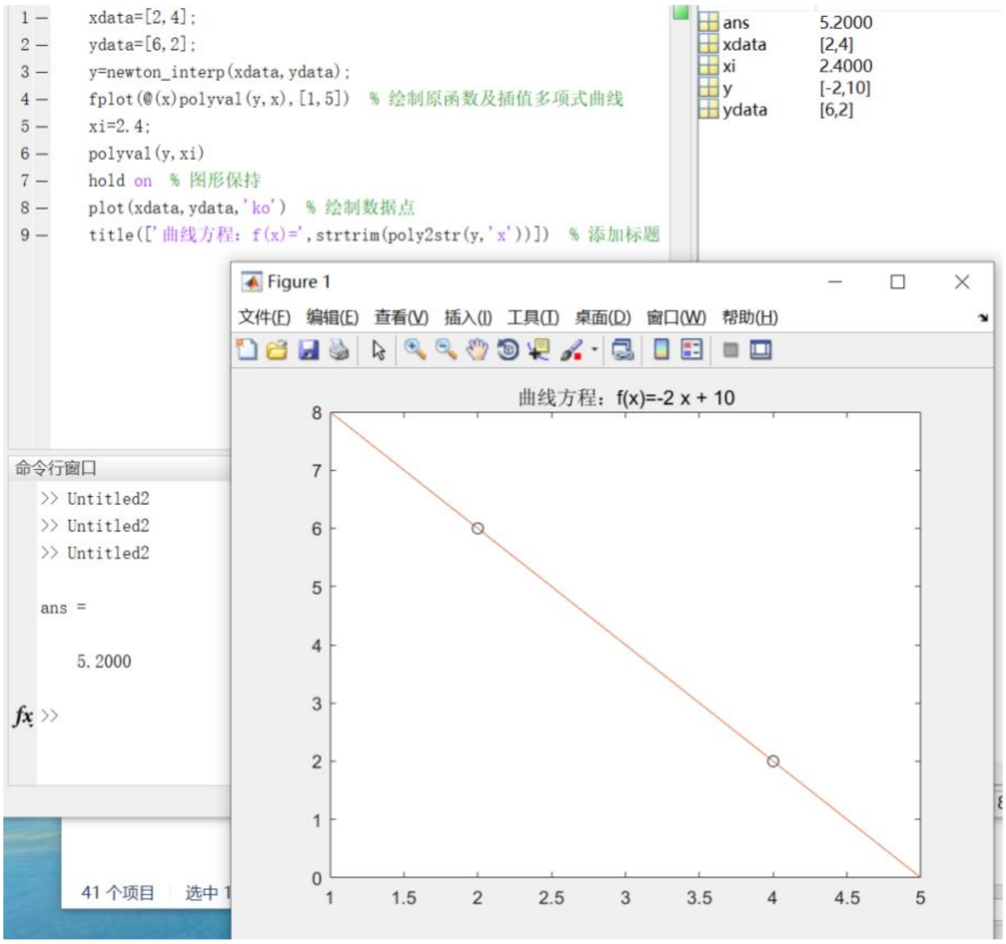
$$\therefore \boldsymbol{f(2.4) \approx N(2.4)} = 5.2$$

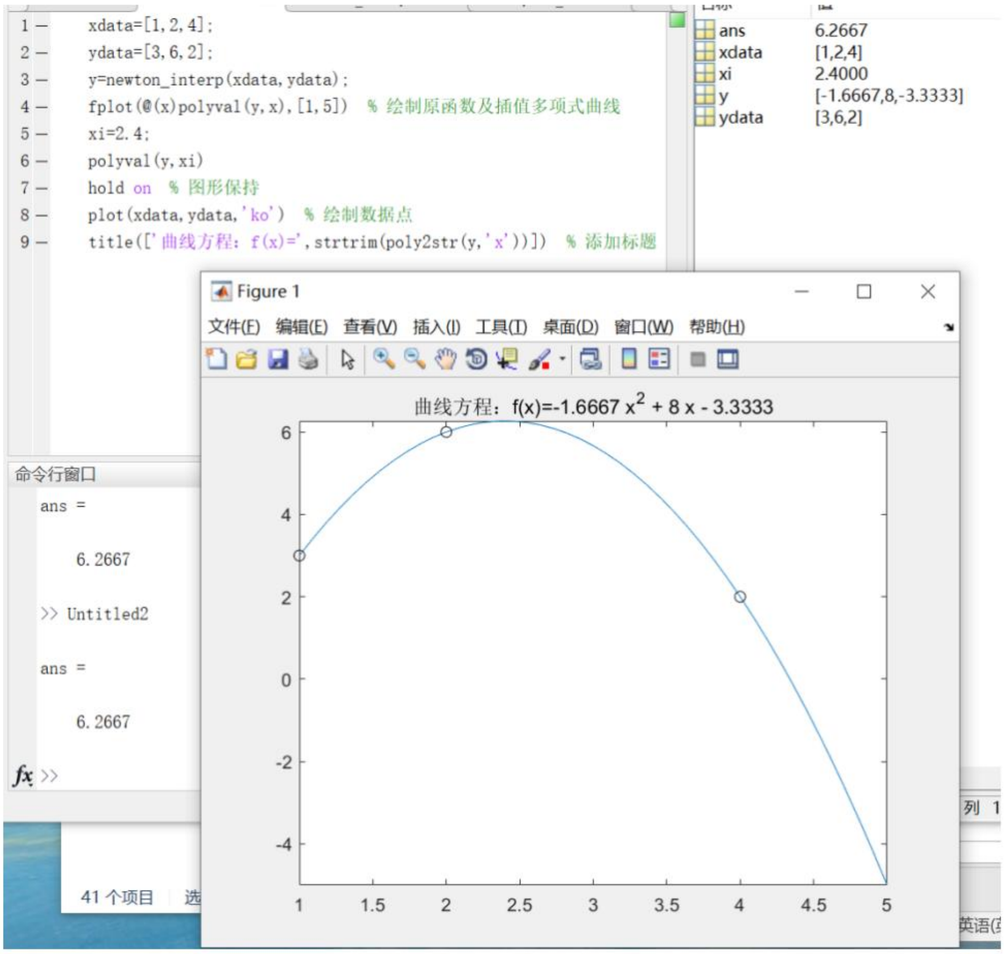
用二次Newton插值近似计算*f(2.4)*，应选与2.4最近的3个节点：

$$x_0 = 1, x_1 = 2, x_2 = 4$$

由表中 行数据有 $N_2(x) = 3 + 3(x-1) - \frac{5}{3}(x-1)(x-2) = -\frac{10}{3} + 8x - \frac{5}{3}x^2$

$$\therefore f(2.4) \approx N_2(2.4) = 6.26667$$




xdata=[1,2,4];
ydata=[3,6,2];
y=newton_interp(xdata,ydata);
fplot(@(x)polyval(y,x),[1,5]) % 绘制原函数及插值多项式曲线
xi=2.4;
polyval(y,xi)
hold on % 图形保持
plot(xdata,ydata,'ko') % 绘制数据点
title(['曲线方程：f(x)=',strtrim(poly2str(y,'x'))]) % 添加标题
ans 6.2667
xdata [1,2,4]
xi 2.4000
y [-1.6667,8,-3.3333]
ydata [3,6,2]
命令行窗口
ans =
6.2667
>> Untitled2
ans =
6.2667
Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
曲线方程：f(x)=-1.6667 x^2 + 8 x - 3.3333

【实验内容】

实验要求：

任务1.给定数表

x	1	2	4	5
y	3	6	2	1

1.给出差商表;

2.试用一次和二次Newton插值多项式计算 $f(2.4)$ 的近似值.

解: 1.差商表;

x	y	$f[?,?]$	$f[?,?,?]$	$f[?,?,?,?]$
1	3	3	$-5/3$	$1/2$
2	6	-2	$1/3$	
4	2	-1		
5	1			

```
xdata=[1 2 4 5];
ydata=[3 6 2 1];
d=diffquot(xdata,ydata)
```

命令行窗口

```
>> example6_3

d =

    1.0000    3.0000    3.0000   -1.6667    0.5000
    2.0000    6.0000   -2.0000    0.3333         0
    4.0000    2.0000   -1.0000         0         0
    5.0000    1.0000         0         0         0

>>
```

实验代码:

code1_yyz.m

```
% 任务 1
% 实验人：杨跃浙
xdata = [1, 2, 4, 5]; % x 数据
ydata = [3, 6, 2, 1]; % y 数据
x = 2.4; % 要插值的点

% 调用 Newton 插值函数
[result1, result2, diff_table] = newton_yyz(xdata, ydata, x);

% 获取差商表的大小
[num_rows, num_cols] = size(diff_table);

% 打印差商表（加表头）
fprintf('差商表:\n');
fprintf('%10s', 'x_i'); % 打印第一列标题
fprintf('%10s', 'f(x_i)'); % 打印第二列标题
for order = 1:(num_cols - 1)
fprintf('%10s', sprintf('Δ^%d', order)); % 动态生成差商列标题
end
fprintf('\n');

% 打印差商表数据
for i = 1:num_rows
```

```
fprintf('%10.4f', xdata(i)); % 打印当前的 x_i 值
fprintf('%10.4f', diff_table(i, 1)); % 打印 f(x_i) 值，从 diff_table 获取
for order = 2:num_cols
if (order - 1) <= (num_rows - i)
val = diff_table(i, order);
% 检查差商值是否为空或 NaN，若是则填充 0
if isempty(val) || isnan(val)
val = 0;
end
fprintf('%10.4f', val);
else
fprintf('%10.4f', 0); % 填充 0 以保持对齐
end
end
fprintf('\n');
end

% 输出插值结果
fprintf('一次插值多项式在 x = %.2f 处的近似值为: %.4f\n', x, result1);
fprintf('二次插值多项式在 x = %.2f 处的近似值为: %.4f\n', x, result2);
```

newton_yyz.m

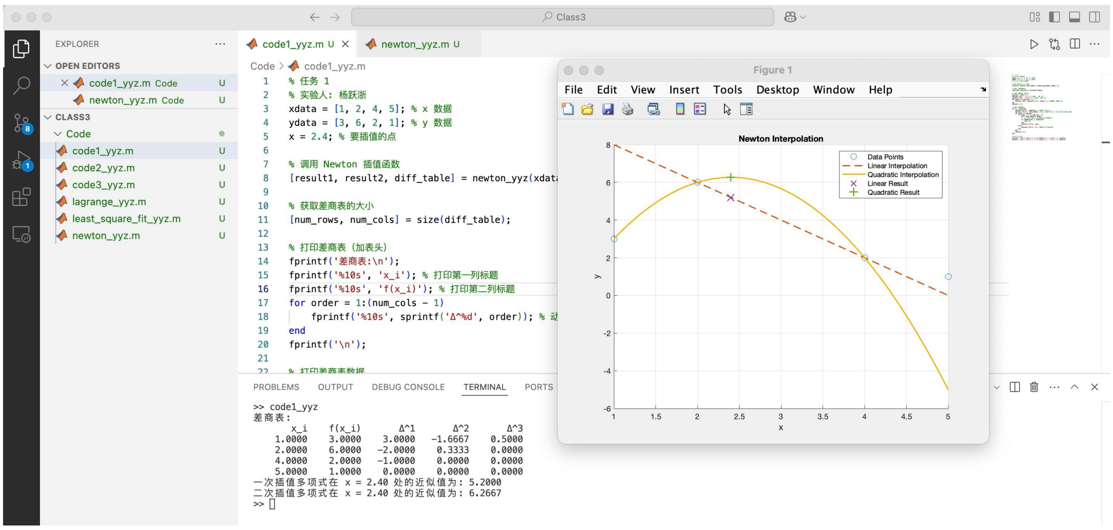
```
function [result1, result2, diff_table] = newton_yyz(xdata, ydata, x)
```

```
n = length(xdata); % 数据点数量
diff_table = zeros(n, n); % 差商表
diff_table(:, 1) = ydata'; % 第一列为 ydata

% 计算差商表
for j = 2:n
for i = 1:(n - j + 1)
diff_table(i, j) = ...
(diff_table(i + 1, j - 1) - diff_table(i, j - 1)) ...
/ (xdata(i + j - 1) - xdata(i));
end
end

[~, idx] = sort(abs(xdata - x)); % 按距离 x 的绝对值排序
x_selected = xdata(idx); % 最近点的 x 数据
y_selected = ydata(idx); % 最近点的 y 数据
x0 = x_selected(1); % 最近的 x0
x1 = x_selected(2); % 最近的 x1
f_x0 = y_selected(1); % 对应的 f(x0)
delta1 = diff_table(idx(1), 2); % Δ^1(x0)

% 一次插值公式
result1 = f_x0 + delta1 * (x - x0);
```

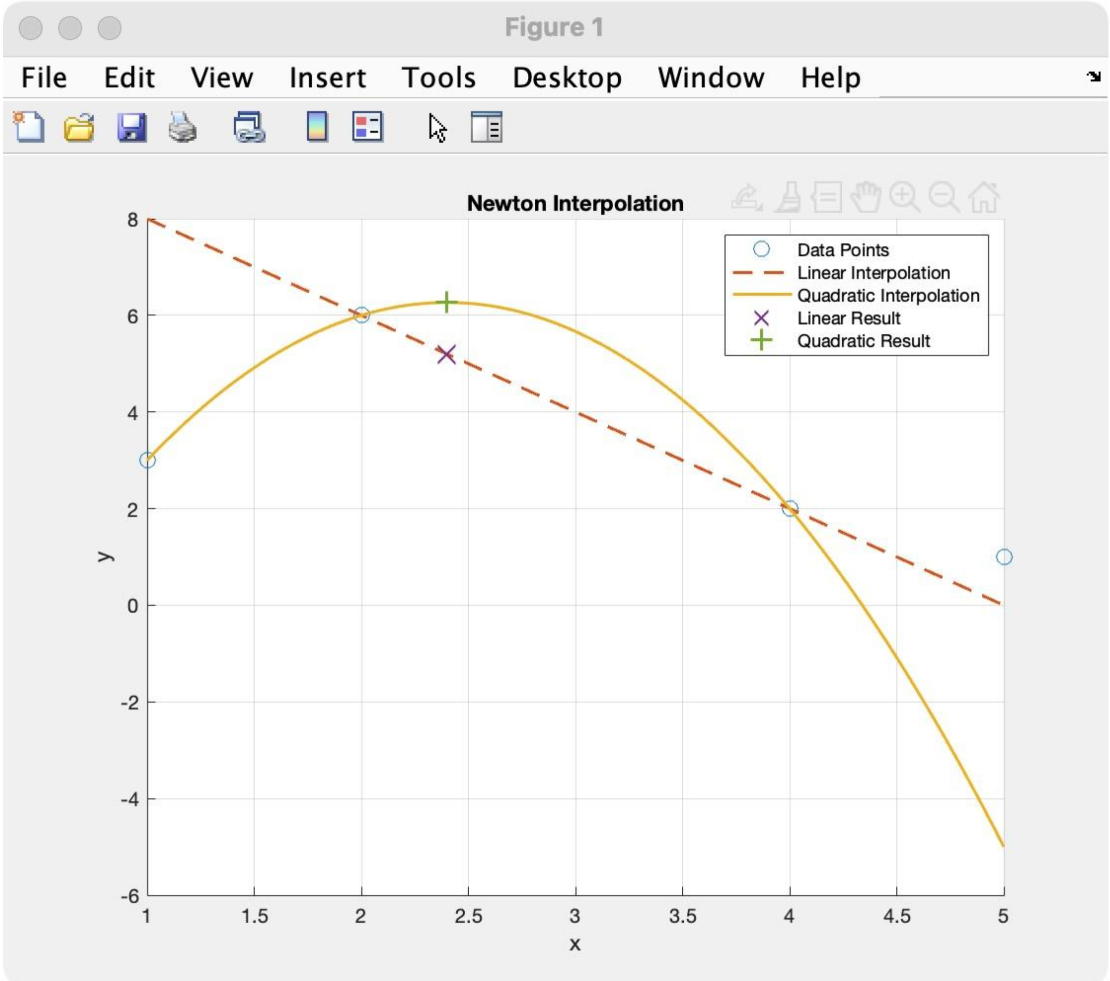
```
delta2 = diff_table(x1, 3); % Δ^2(x1)
f_x1 = y_selected(2); % 对应的 f(x1)
delta11 = diff_table(x1, 2); % Δ^1(x1)
% 二次插值公式
result2 = f_x1 + delta11 * (x - x1) + delta2 * (x - x0) * (x - x1);

% 画图
% 定义细化的 x 值用于绘制插值曲线
x_fine = linspace(min(xdata), max(xdata), 100);
y_fine1 = f_x0 + delta1 * (x_fine - x0); % 一次插值曲线
y_fine2 = f_x1 + delta11 * (x_fine - x1) + delta2 * (x_fine - x0) .* (x_fine - x1); % 二次插值曲线

figure;
hold on;
% 原始数据点
plot(xdata, ydata, 'o', 'MarkerSize', 8, 'DisplayName', 'Data Points');
% 一次插值曲线
plot(x_fine, y_fine1, '--', 'LineWidth', 1.5, 'DisplayName', 'Linear Interpolation');
% 二次插值曲线
plot(x_fine, y_fine2, '-', 'LineWidth', 1.5, 'DisplayName', 'Quadratic Interpolation');
```

```matlab
% 标记输入点
plot(x, result1, 'x', 'MarkerSize', 10, 'LineWidth', 1.5, 'DisplayName', 'Linear Result');
plot(x, result2, '+', 'MarkerSize', 10, 'LineWidth', 1.5, 'DisplayName', 'Quadratic Result');
legend('show');
title('Newton Interpolation');
xlabel('x');
ylabel('y');
grid on;
hold off;
end
```

运行截图：



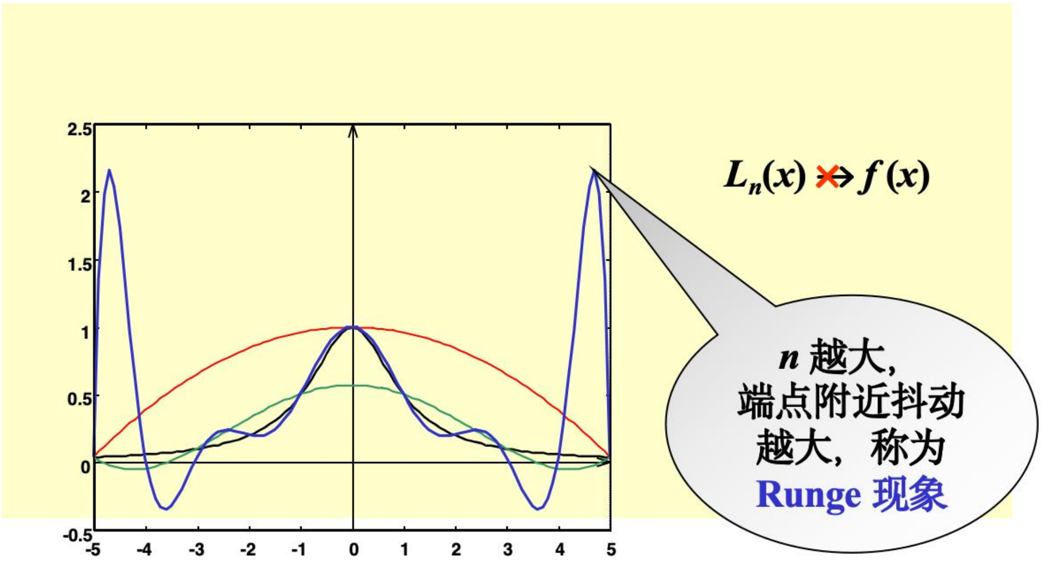
```
>> code1_yyz
差商表：
        x_i      f(x_i)        Δ^1        Δ^2        Δ^3
    1.0000     3.0000     3.0000    -1.6667     0.5000
    2.0000     6.0000    -2.0000     0.3333     0.0000
    4.0000     2.0000    -1.0000     0.0000     0.0000
    5.0000     1.0000     0.0000     0.0000     0.0000
一次插值多项式在 x = 2.40 处的近似值为： 5.2000
二次插值多项式在 x = 2.40 处的近似值为： 6.2667
```



实验要求：

任务2：在$[-5, 5]$上考察$f(x)=\dfrac{1}{1+x^2}$的$L_n(x)$。取$x_i=-5+\dfrac{10}{n}i\quad(i=0,\ldots,n)$

验证Runge现象



实验代码：

code2_yyz.m

```
% 任务 2
% 实验人：杨跃浙


f = @(x) 1 ./ (1 + x.^2);

% 定义区间
a = -5; % 区间左端点
b = 5; % 区间右端点

% 用于绘图的细化点
x_fine = linspace(a, b, 1000);
```

```
y_fine = f(x_fine);

% 不同阶数的插值
n_values = [1, 2, 3, 4, 5, 6, 8, 10]; % 插值点个数
colors = lines(length(n_values)); % 不同阶数的颜色
figure;
hold on;

% 绘制原函数
plot(x_fine, y_fine, 'k-', 'LineWidth', 1.5, 'DisplayName', 'f(x) =
1/(1+x^2)');

% 遍历不同阶数
for k = 1:length(n_values)
n = n_values(k);
% 等距插值点
x_nodes = linspace(a, b, n + 1);
y_nodes = f(x_nodes);

% 调用拉格朗日插值函数
y_interp = lagrange_yyz(x_nodes, y_nodes, x_fine);

% 绘制插值曲线
plot(x_fine, y_interp, '-', 'LineWidth', 1.2, 'Color', colors(k, :), ...
'DisplayName', sprintf('L_{%d}(x)', n));
```

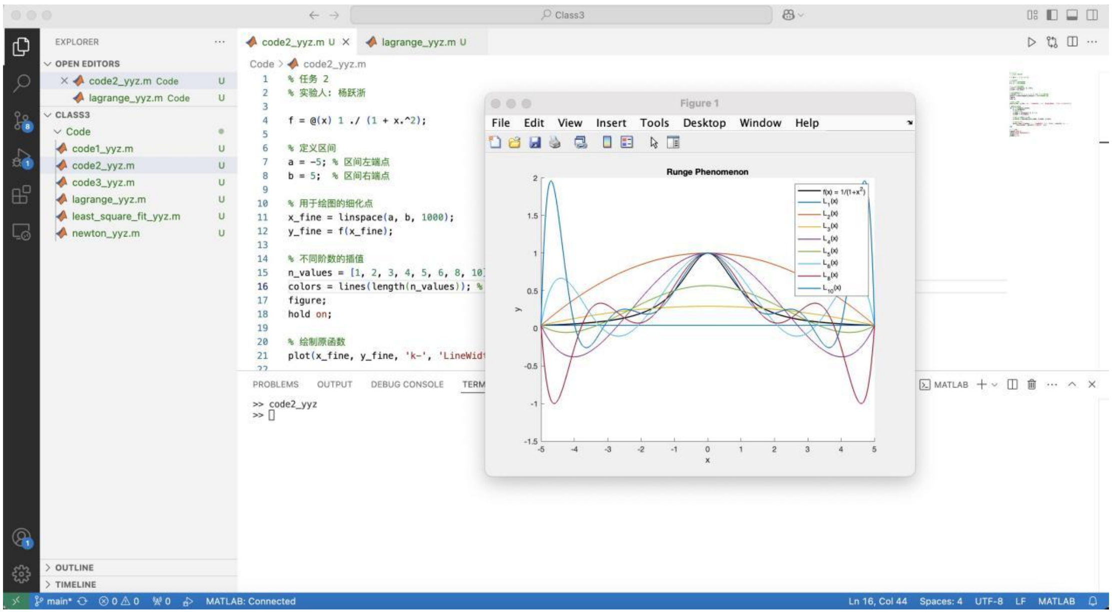
```
end

% 图形美化
legend('show');
title('Runge Phenomenon');
xlabel('x');
ylabel('y');
```

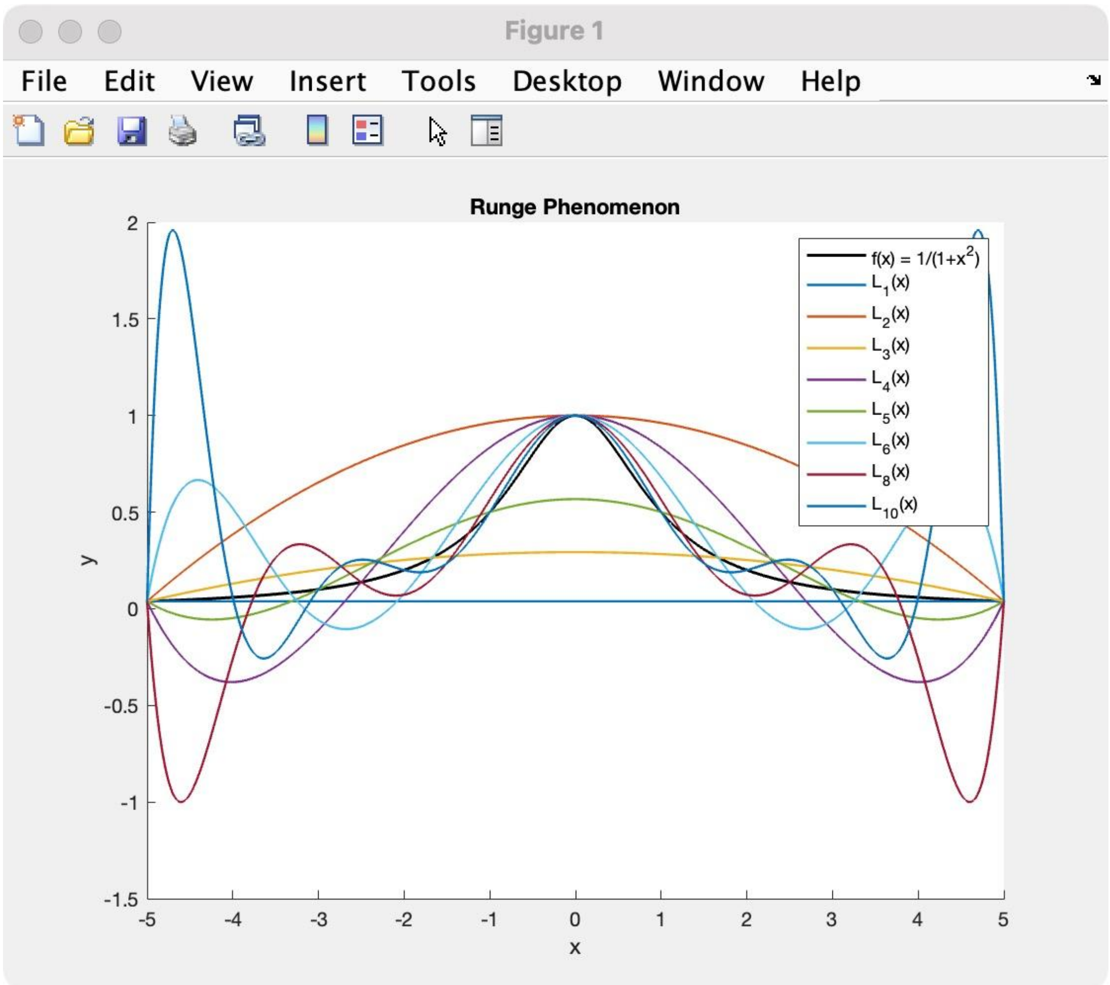
lagrange_yyz.m

```
function y_interp = lagrange_yyz(x_nodes, y_nodes, x_fine)
y_interp = zeros(size(x_fine));

% 遍历每个节点，计算拉格朗日基函数并累加
for i = 1:length(x_nodes)
% 初始化基函数 L_i(x)
L = ones(size(x_fine));
for j = 1:length(x_nodes)
if j ~= i
L = L .* (x_fine - x_nodes(j)) / (x_nodes(i) - x_nodes(j));
end
end
% 累加插值值
y_interp = y_interp + y_nodes(i) * L;
end
```

```
end
```

运行截图：





实验要求：

任务3: 给定数据表

x	-3	-1	0	1	3
y	-1.2	1.3	1.5	1.9	2

求二次拟合曲线.

解： 求二次拟合曲线就是求2次最小二乘多项式，由于没给出权系数，可选权为1.所求拟合函数类型为

$$\varphi(x) = a_0 + a_1 x + a_2 x^2$$

$$\therefore \varphi_0(x) = 1, \varphi_1(x) = x, \varphi_2(x) = x^2$$

$$\begin{bmatrix} \sum_{i=1}^{N}\omega_i & \sum_{i=1}^{N}\omega_i x_i & \cdots & \sum_{i=1}^{N}\omega_i x_i^m \\ \sum_{i=1}^{N}\omega_i x_i & \sum_{i=1}^{N}\omega_i x_i^2 & \cdots & \sum_{i=1}^{N}\omega_i x_i^{m+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{N}\omega_i x_i^m & \sum_{i=1}^{N}\omega_i x_i^{m+1} & \cdots & \sum_{i=1}^{N}\omega_i x_i^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N}\omega_i y_i \\ \sum_{i=1}^{N}\omega_i x_i y_i \\ \vdots \\ \sum_{i=1}^{N}\omega_i x_i^m y_i \end{bmatrix}$$

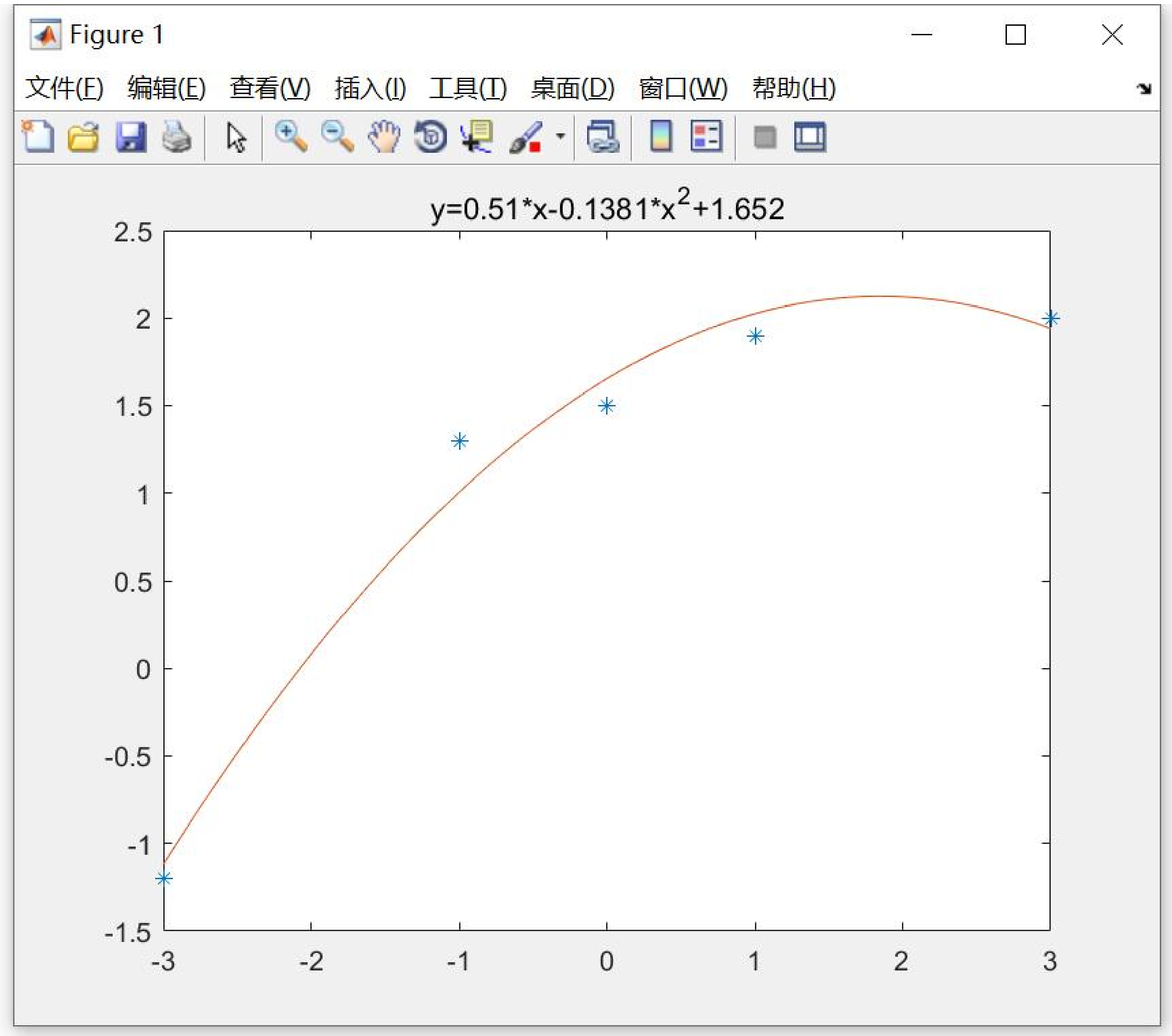
N=5，m=2，算得法方程组

$$\begin{bmatrix} 5 & 0 & 20 \\ 0 & 20 & 0 \\ 20 & 0 & 164 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 5.5 \\ 10.2 \\ 10.4 \end{bmatrix}$$

解之得 $a_0^* = 1.6524, a_1^* = 0.51, a_2^* = -0.1381$

$$\therefore \varphi^*(x) = 1.6524 + 0.51x - 0.1381x^2$$

$$\begin{bmatrix} \sum_{i=1}^{N}\omega_i & \sum_{i=1}^{N}\omega_i x_i & \cdots & \sum_{i=1}^{N}\omega_i x_i^m \\ \sum_{i=1}^{N}\omega_i x_i & \sum_{i=1}^{N}\omega_i x_i^2 & \cdots & \sum_{i=1}^{N}\omega_i x_i^{m+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{N}\omega_i x_i^m & \sum_{i=1}^{N}\omega_i x_i^{m+1} & \cdots & \sum_{i=1}^{N}\omega_i x_i^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N}\omega_i y_i \\ \sum_{i=1}^{N}\omega_i x_i y_i \\ \vdots \\ \sum_{i=1}^{N}\omega_i x_i^m y_i \end{bmatrix}$$



实验代码：

code3_yyz.m

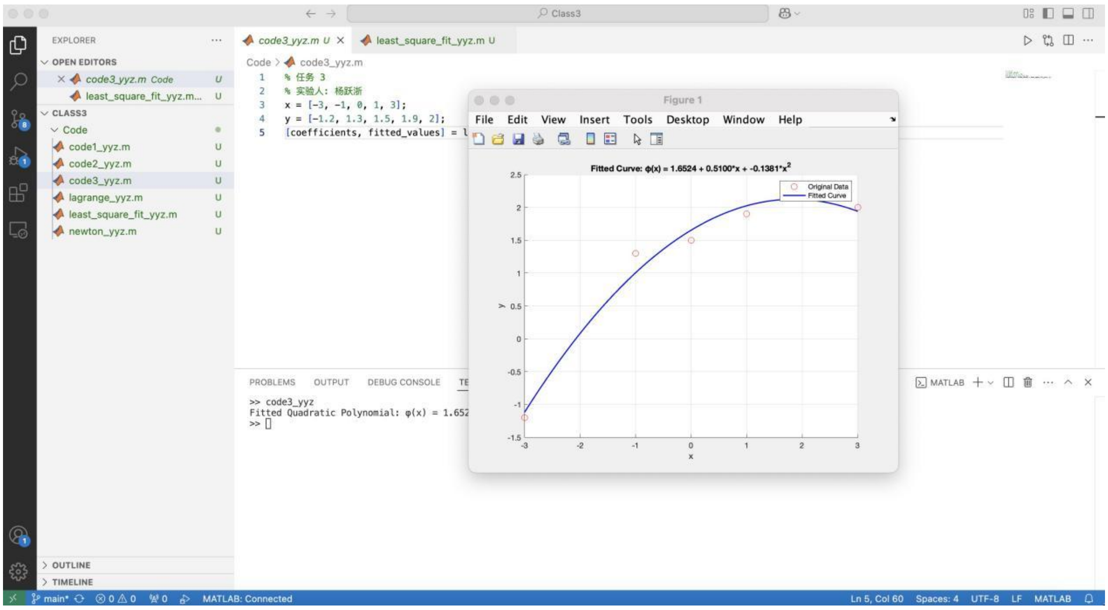
```
% 任务 3

% 实验人：杨跃浙

x = [-3, -1, 0, 1, 3];

y = [-1.2, 1.3, 1.5, 1.9, 2];

[coefficients, fitted_values] = least_square_fit_yyz(x, y);
```

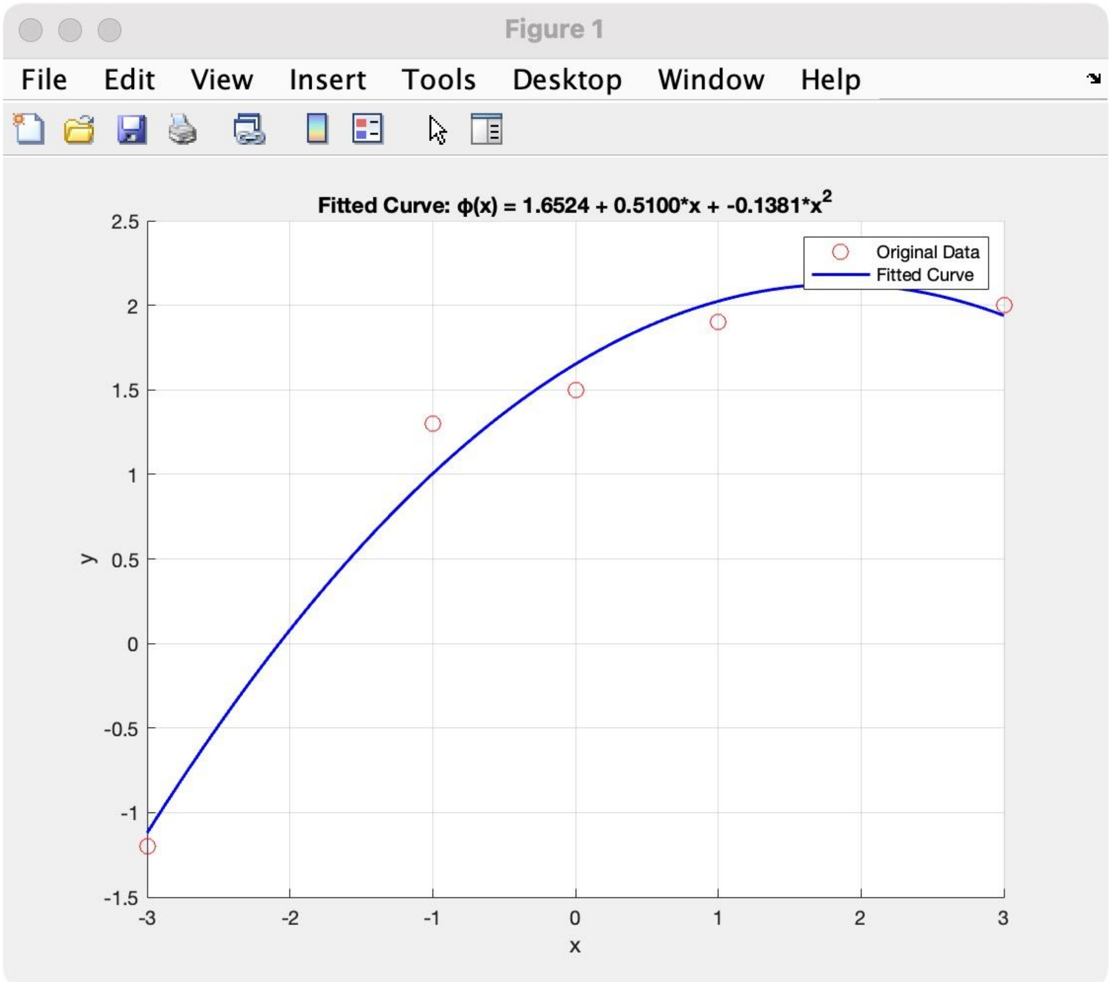
least_square_fit_yyz.m

```
function [coefficients, fitted_values] = least_square_fit_yyz(x, y)
% 构造矩阵 X 和向量 Y
X = [ones(length(x), 1), x(:), x(:).^2]; % [1, x, x^2]
Y = y(:); % 转为列向量
% 最小二乘解
coefficients = (X' * X) \ (X' * Y);
% 计算拟合值
fitted_values = X * coefficients;
% 构造拟合多项式表达式
a0 = coefficients(1);
a1 = coefficients(2);
a2 = coefficients(3);
fprintf('Fitted Quadratic Polynomial: ϕ(x) = %.4f + %.4f*x
+ %.4f*x^2\n', a0, a1, a2);
% 用于绘图的拟合曲线
x_fine = linspace(min(x), max(x), 100); % 细化点
y_fine = a0 + a1 * x_fine + a2 * x_fine.^2;
% 绘制原始数据点和拟合曲线
figure;
hold on;
% 原始数据点
plot(x, y, 'ro', 'MarkerSize', 8, 'DisplayName', 'Original Data');
```

```
% 拟合曲线
plot(x_fine, y_fine, 'b-', 'LineWidth', 1.5, 'DisplayName', 'Fitted Curve');
legend('show');
title(sprintf('Fitted Curve: ϕ(x) = %.4f + %.4f*x + %.4f*x^2', a0, a1, a2));
xlabel('x');
ylabel('y');
grid on;
hold off;
end
```

运行截图：





实验要求：

练习理论课 ppt 及参考资料中介绍的各种算法及案例

案例一：

例 **已知的函数值如下：**

x	1	2	4	5
$f(x)$	1	3	4	2

在区间[1,5]上求三次样条插值函数$S(x)$,使它满足边界条件

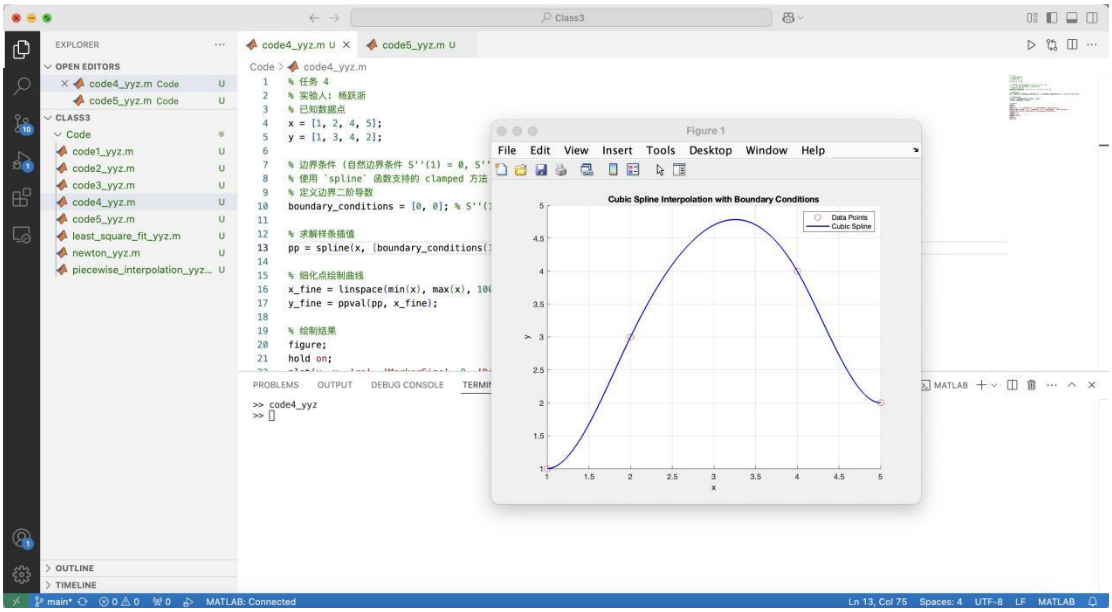
$$S''(1)=0, S''(5)=0$$

实验代码：

```
% 任务 4
% 实验人：杨跃浙
% 已知数据点
x = [1, 2, 4, 5];
y = [1, 3, 4, 2];

% 边界条件（自然边界条件 S''(1) = 0, S''(5) = 0）
% 使用 `spline` 函数支持的 clamped 方法
% 定义边界二阶导数
boundary_conditions = [0, 0]; % S''(1) = 0, S''(5) = 0

% 求解样条插值
pp = spline(x, [boundary_conditions(1), y,
boundary_conditions(2)]); % piecewise polynomial

% 细化点绘制曲线
x_fine = linspace(min(x), max(x), 1000);
y_fine = ppval(pp, x_fine);
```

```
% 绘制结果
figure;
hold on;
plot(x, y, 'ro', 'MarkerSize', 8, 'DisplayName', 'Data Points');
plot(x_fine, y_fine, 'b-', 'LineWidth', 1.5, 'DisplayName', 'Cubic Spline');
title('Cubic Spline Interpolation with Boundary Conditions');
xlabel('x');
ylabel('y');
legend('show');
grid on;
hold off;
```

运行截图：


EXPLORER
OPEN EDITORS
code4_yyz.m
code5_yyz.m
CLASS3
Code
code1_yyz.m
code2_yyz.m
code3_yyz.m
code4_yyz.m
code5_yyz.m
least_square_fit_yyz.m
newton_yyz.m
% 任务 4
% 实验人：杨跃浙
% 已知数据点
x = [1, 2, 4, 5];
y = [1, 3, 4, 2];
boundary_conditions = [0, 0];
figure;
hold on;
Figure 1
File Edit View Insert Tools Desktop Window Help
Cubic Spline Interpolation with Boundary Conditions
PROBLEMS OUTPUT DEBUG CONSOLE
>> code4_yyz
MATLAB: Connected

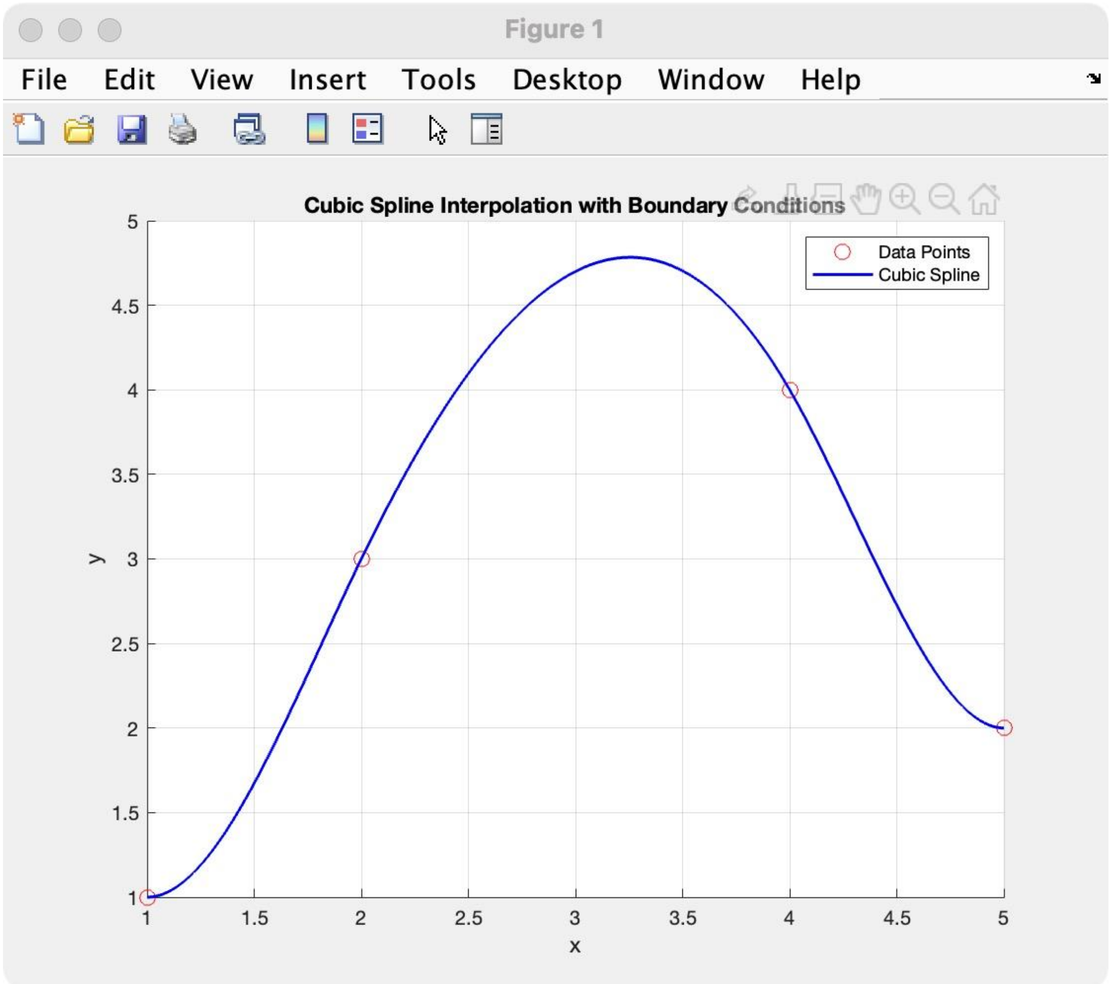


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Cubic Spline Interpolation with Boundary Conditions
Data Points
Cubic Spline
y
x
1
1.5
2
2.5
3
3.5
4
4.5
5

案例二：

试对下列数据做出型如 $a+bx^2$ 的拟合曲线。

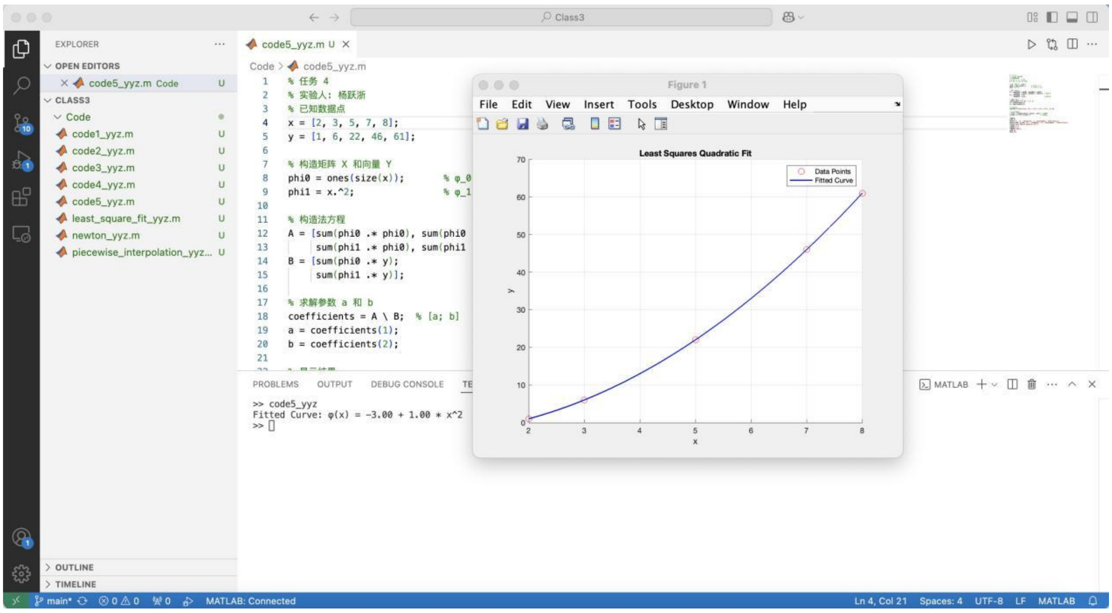
x	2	3	5	7	8
y	1	6	22	46	61

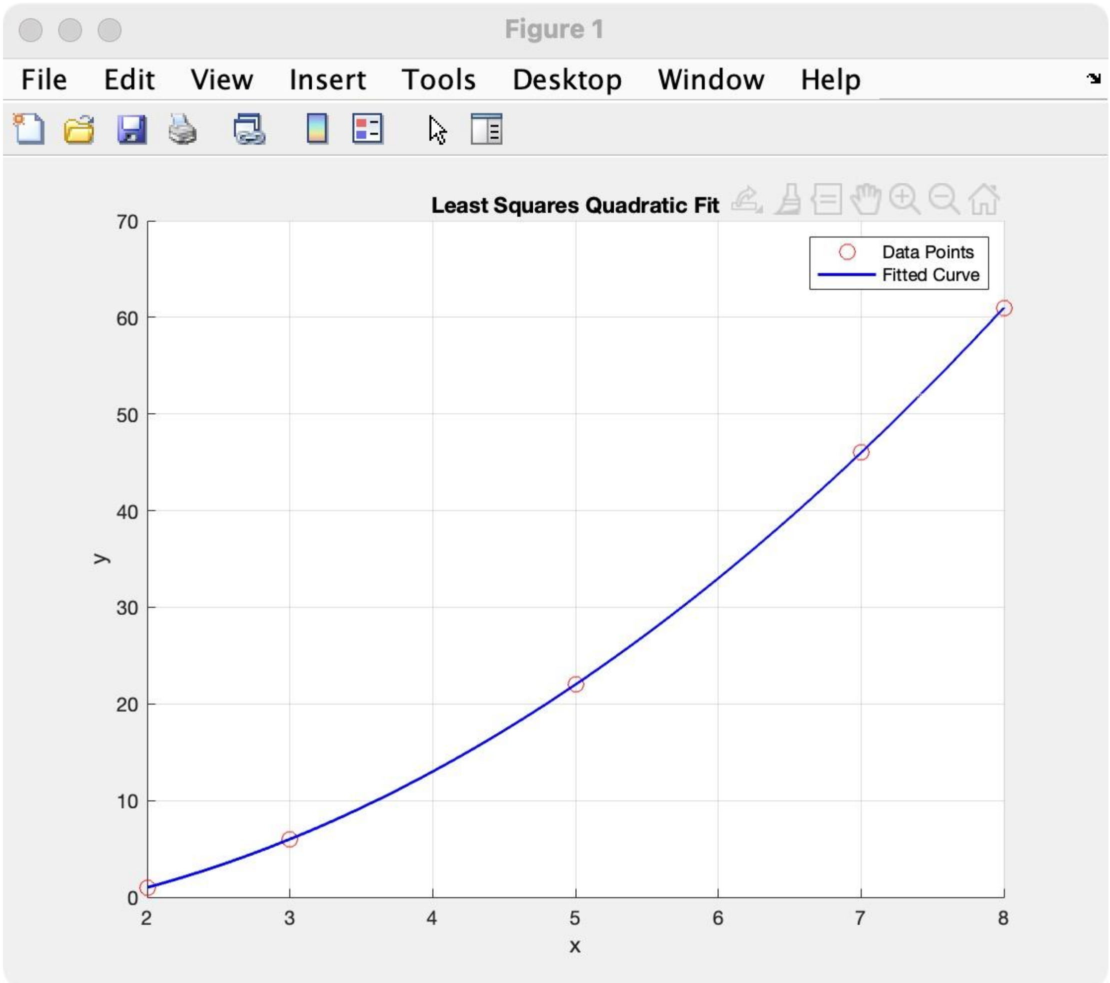
实验代码：

```
% 任务 4

% 实验人：杨跃浙

% 已知数据点

x = [2, 3, 5, 7, 8];

y = [1, 6, 22, 46, 61];


% 构造矩阵 X 和向量 Y

phi0 = ones(size(x)); % ϕ_0(x) = 1

phi1 = x.^2; % ϕ_1(x) = x^2

% 构造法方程

A = [sum(phi0 .* phi0), sum(phi0 .* phi1);

sum(phi1 .* phi0), sum(phi1 .* phi1)]; % 矩阵 A

B = [sum(phi0 .* y);

sum(phi1 .* y)]; % 矩阵 B

% 求解参数 a 和 b

coefficients = A \ B; % [a; b]

a = coefficients(1);
```

```
b = coefficients(2);

% 显示结果
fprintf('Fitted Curve: ϕ(x) = %.2f + %.2f * x^2\n', a, b);

% 绘制拟合曲线
x_fine = linspace(min(x), max(x), 1000); % 细化点
y_fine = a + b * x_fine.^2; % 拟合曲线

figure;
hold on;
plot(x, y, 'ro', 'MarkerSize', 8, 'DisplayName', 'Data Points');
plot(x_fine, y_fine, 'b-', 'LineWidth', 1.5, 'DisplayName', 'Fitted
Curve');
title('Least Squares Quadratic Fit');
xlabel('x');
ylabel('y');
legend('show');
grid on;
hold off;
```

运行截图：





【小结或讨论】

在本次实验中，我们深入探究了多种数值计算方法，包括插值法

和曲线拟合方法，取得了一系列有意义的结果，并对其进行了深入分析。

一、插值法

（一）Newton 插值法

在处理给定数表数据时，Newton 插值法展现出了其独特的优势。通过构建差商表，我们能够系统地计算出各阶差商，为插值多项式的构建提供了关键数据。以计算 $f(2.4)$ 为例，一次插值多项式 $N_1(x)=f(x_0)+\Delta^1(x_0)(x-x_0)$ ，其中 $\Delta^1(x_0)=\frac{f(x_1)-f(x_0)}{x_1-x_0}$ 。在本题中，选取 $x_0=2, x_1=4$ ，计算可得 $\Delta^1(2)=\frac{2-6}{4-2}=-2$ ，则 $N_1(x)=6-2(x-2)=-2x+10$ ，进而求得 $f(2.4)\approx N_1(2.4)=5.2$.

二次插值多项式 $N_2(x)=f(x_1)+\Delta^1(x_1)(x-x_1)+\Delta^2(x_1)(x-x_0)(x-x_1)$ ，其中 $\Delta^2(x_1)=\frac{\Delta^1(x_2)-\Delta^1(x_1)}{x_2-x_1}$ 。选取 $x_0=1, x_1=2, x_2=4$，经计算 $\Delta^1(1)=\frac{6-3}{2-1}=3$ ， $\Delta^1(2)=-2, \Delta^2(2)=\frac{-2-3}{4-1}=-\frac{5}{3}$ ，所以

$$N_2(x)=3+3(x-1)-\frac{5}{3}(x-1)(x-2)$$

$$=-\frac{10}{3}+8x-\frac{5}{3}x^2, f(2.4)\approx N_2(2.4)=6.26667$$

从结果来看，二次插值相较于一次插值在一定程度上提高了近似值的精度，这表明增加插值节点数量和提高插值多项式阶数可以改善逼近效果，但同时也可能增加计算复杂度和误差传播风险。

（二）Lagrange 插值法与 Runge 现象

对于函数 $f(x)=\frac{1}{1+x^2}$ 在区间 $[-5,5]$ 上的研究，我们运用 Lagrange 插值法进行了不同节点数量下的插值计算。随着 n 的增大，在区间端点附近出现了明显的抖动现象，即 Runge 现象。这是因为高次插值多项式在某些情况下会出现振荡，导致插值结果偏离原函数。

从理论上讲，Lagrange 插值多项式为 $L_n(x)=\sum_{i=0}^{n} f(x_i)\frac{\prod_{j=0,j\neq i}^{n}(x-x_j)}{\prod_{j=0,j\neq i}^{n}(x_i-x_j)}$。在实际计算中，我们发现虽然增加节点可以使插值多项式在更多点上与原函数值相等，但在端点附近却可能产生较大误差。这提示我们在实际应用中，不能盲目追求高次插值，需要根据数据特点和精度要求合理选择插值方法和节点数量。

二、曲线拟合

（一）二次拟合曲线

在求解给定数据表的二次拟合曲线 $\phi(x)=a_0+a_1x+a_2x^2$

时，我们通过构建法方程组

$$\begin{bmatrix} \sum_{i=1}^{N} \omega_i & \sum_{i=1}^{N} \omega_i x_i & \sum_{i=1}^{N} \omega_i x_i^2 \\ \sum_{i=1}^{N} \omega_i x_i & \sum_{i=1}^{N} \omega_i x_i^2 & \sum_{i=1}^{N} \omega_i x_i^3 \\ \sum_{i=1}^{N} \omega_i x_i^2 & \sum_{i=1}^{N} \omega_i x_i^3 & \sum_{i=1}^{N} \omega_i x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} \omega_i y_i \\ \sum_{i=1}^{N} \omega_i x_i y_i \\ \sum_{i=1}^{N} \omega_i x_i^2 y_i \end{bmatrix} \text{(本题中权系数 } \omega_i = 1)$$

，并求解得到 $a_0^* = 1.6524, a_1^* = 0.51, a_2^* = -0.1381$,

即 $\phi^*(x) = 1.6524 + 0.51x - 0.1381x^2$ 。

通过绘制原始数据点和拟合曲线，可以直观地看到拟合曲线较好地反映了数据的大致趋势，但在某些点上仍存在一定偏差。这可能是由于数据本身的噪声或所选拟合函数形式的局限性所致。在实际应用中，我们可以尝试不同的拟合函数形式或增加数据量来提高拟合精度。

（二）特定形式拟合曲线

对于型如 $a + bx^2$ 的拟合曲线，同样通过构建法方程组求解参

数。根据给定数据计算得到相应的矩阵元素，进而求得参数 a 和 b 的值，确定拟合曲线表达式。绘制图形后发现，该拟合曲线在一定程度上捕捉到了数据的变化规律，但与二次拟合曲线类似，也存在一定的拟合误差。这说明在选择拟合曲线形式时，需要充分考虑数据的特点和潜在的函数关系，以获得更优的拟合效果。

三、三次样条插值

在求解三次样条插值函数 S(x) 时，利用 spline 函数结合给定边界条件成功得到了插值结果。三次样条插值在保证曲线光滑性方面表现出色，能够较好地拟合给定数据点，并且在区间内的变化较为自然。与其他插值方法相比，它在处理具有复杂变化趋势的数据时具有明显优势，能够有效避免高次插值的振荡问题，为数据的精确插值提供了一种可靠的方法。

通过本次实验，我们不仅掌握了多种数值计算方法的原理和应用技巧，还深刻认识到不同方法在不同场景下的优缺点。在实际问题中，我们需要根据数据的特点，精度要求和计算资源等因素综合选择合适的方法，以获得最满意的结果。同时，对于实验中出现的误差和问题，我们需要进一步深入分析，不断改进和优化计算过程，提高数值计算的准确性和可靠性。