实验序号： 实验 3 **实验项目：** 插值法和最小二乘拟合法

实验成绩： ________ **教师签名：** ________

西 南 大 学 人 工 智 能 学 院
实 验 报 告

学年学期	2023 年秋季学期
课程名称	数值分析
姓　　名	洪浩钦
学　　号	222021335210144
学　　院	含弘学院
专　　业	智能科学与技术
班　　级	袁隆平班
任课教师	**钟秀蓉**

2023 年 **12** 月 **13** 日

一、实验目的

1. 了解不同插值方法(拉格朗日插值法、牛顿插值法)在实现和原理上的区别。
2. 掌握拉格朗日插值法和牛顿插值法在 MATLAB 中的实现方法。
3. 比较拉格朗日插值法和牛顿插值法在不同阶数下的拟合效果。
4. 了解最小二乘法在线性拟合中的应用,并掌握实现最小二乘法的步骤。
5. 分析不同插值算法和最小二乘法在拟合不同数据分布下的优劣。

二、算法原理概述

1.拉格朗日插值法

拉格朗日插值法是一种多项式插值方法。它根据给定的数据点(xi,yi),利用拉格朗日基函数 li(x)定义唯一的插值多项式:

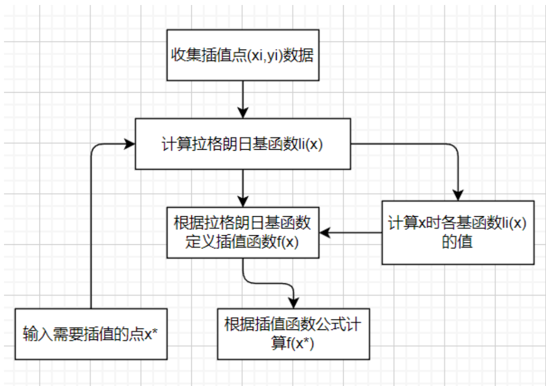
$$f(x) = \sum yi*li(x)$$

其中,拉格朗日基函数定义为:

$$li(x) = \prod(x - xj)/(xi - xj),\ i \neq j$$

该插值多项式可以完全逼近数据点,阶数为 n-1。拉格朗日插值法算法简单,在数据点周围更平滑,是多项式插值的基础方法之一。

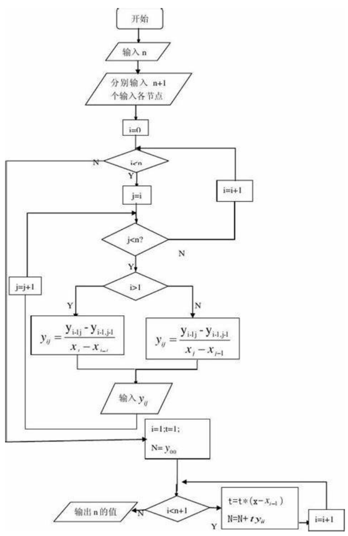
拉格朗日插值法的计算流程如右图。



2.牛顿插值法

牛顿插值法是一种基于差商的多项式插值方法。它利用差商描述函数在区间内的泰勉展开近似,将插值式展开为多项式形式,考虑更高阶导数信息获得更高精度。但需要更多数据点来计算差商,计算量大。它在数据点附近更准确,但随距离增大误差会增大。相比拉格朗日插值,牛顿插值法插值精度高,但假设和计算难度大。

牛顿插值法的计算流程如右图。



3.多项式拟合

[1] 输入数据:接收输入数据 x,y 以及拟合多项式的阶数 m。

[2] 构建增广矩阵:构建大小为(m+1)×(m+1)的矩阵 A,保存多项式各项的和。同时构建大小为(m+1)×1 的向量 b,保存 x 的各次幂与 y 的乘积和。

[3] 填充矩阵:使用循环依次计算 A 矩阵和 b 向量的每个元素,分别记录多项式各项

的和和各项与 y 的乘积和。

[4] 求解参数:使用矩阵反演 A\b 求解线性方程组,得到多项式各元数组成的向量 a。

[5] 返回结果:反转 a 的顺序输出多项式各项参数,实现多项式曲线的拟合。

三、软件开发环境及工具

MATLAB 2021b

四、实验内容

实验 2：利用拉格朗日插值做出 Runge 现象的图像：

设 f(x) = 1/(1+sqrt(x^2))，分别讨论将[-5, 5]区间 5 等分与 10 等分后拉格朗日插值的效果。

1.问题分析：

拉格朗日插值是一种插值方法,它通过多项式插值来得到插值函数。对于给定的点集，拉格朗日插值函数为：

$$L(x) = \sum_{i=1}^{n} y_i l_i(x)$$

Runge 现象是指,对于高次插值,随着插值点的增加,在区间内部误差不断增大,可能会有振荡的现象出现。这是因为高次插值多项式的系数与数据点的位置和数量密切相关。随着数据点的增加,高次项的影响增大,可能导致多项式在区间内部处于不稳定状态。

2.实验代码

（1）定义拉格朗日插值函数

```
function yy = nalagr(x, y, xx)
% 用途： Lagrange 插值法数值求解
% 格式： yy =nalagr(x, y, xx)
% x 是节点向量，y是节点上的函数值，xx是插值点，yy返回插值
m = length(x); n = length(y);
if m ~= n, error('向量 x 与 y 的长度必须一致'); end

s = 0;
for i = 1 : n
    t = ones(1, length(xx));
    for j = [1 : i - 1, i + 1 : n]
        t = t .* (xx - x(j)) / (x(i) - x(j)); % 求拉格朗日基函数
```

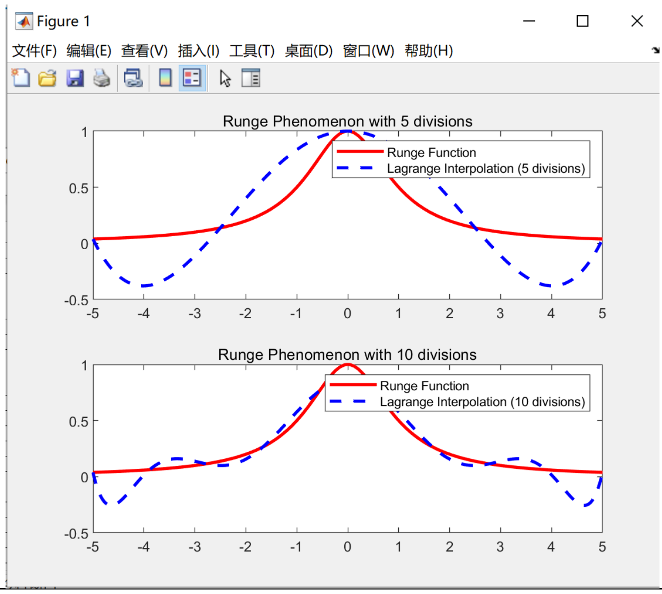
```
        end
        s = s + t * y(i);
end

yy = s;
```

（2）利用拉格朗日插值作出 Runge 现象的图像

```
% 用拉格朗日插值作出 Runge 现象的图像

% 定义 Runge 函数
runge = @(x) 1 ./ (1 + x .^ 2);

% 定义插值点
xx = linspace(-5, 5, 1000);

% 5 等分
x5 = linspace(-5, 5, 5);
y5 = runge(x5);
yy5 = nalagr(x5, y5, xx);

% 10 等分
x10 = linspace(-5, 5, 10);
y10 = runge(x10);
yy10 = nalagr(x10, y10, xx);

% 绘制 Runge 函数和插值函数
figure;
subplot(2,1,1);
plot(xx, runge(xx), 'r-', 'LineWidth', 2); hold on;
plot(xx, yy5, 'b--', 'LineWidth', 2); hold off;
legend('Runge Function', 'Lagrange Interpolation (5 divisions)');
title('Runge Phenomenon with 5 divisions');

subplot(2,1,2);
plot(xx, runge(xx), 'r-', 'LineWidth', 2); hold on;
plot(xx, yy10, 'b--', 'LineWidth', 2); hold off;
legend('Runge Function', 'Lagrange Interpolation (10 divisions)');
title('Runge Phenomenon with 10 divisions');
```

装 订 线

3.实验结果



实验 3：编写牛顿插值法程序，求解习题 9

利用函数 f(x) = sqrt(x) 在 100，121，144 的值，分别用线性插值和抛物插值求 sqrt(115)的近似值。

1.问题分析

牛顿插值法是一种根据已知数据点的值,推算出新点的值的数值插值方法。

主要思想和特点如下:

1. 利用差商来描述函数在数据点附近的变化规律。差商是函数值与自变量相对变化率的比值,用来近似描述函数在区间内的形式。
2. 根据泰勒公式,将插值式展开为多项式形式。泰勒公式可以用差商来近似描述函数在一个区间内的变化。牛顿插值法利用泰勒公式展开到阶数 n 的多项式形式。
3. 具有高精度。与线性插值相比,牛顿插值利用更高阶的项,可以更精细地拟合函数曲线,插值精度高。

2.实验代码

（1）定义牛顿插值函数

```
function yy = newton_interpolation(x, y, xx)
% 用途： Newton 插值法数值求解
% 格式： yy = newton_interpolation(x, y, xx)
% x 是节点向量，y是节点上的函数值，xx是插值点，yy返回插值
n = length(x);
if length(y) ~= n, error('向量 x 与 y 的长度必须一致'); end
```

```
% 计算差分系数表
diff_coeff = zeros(n, n);
diff_coeff(:,1) = y(:); % 第一列是 y

for j = 2:n
    for i = j:n
        diff_coeff(i,j) = (diff_coeff(i,j-1) - diff_coeff(i-1,j-1)) / (x(i) - x(i-j+1)); % 差商
    end
end

% 计算插值
yy = diff_coeff(n, n);
for i = (n-1):-1:1
    yy = yy .* (xx - x(i)) + diff_coeff(i, i);
end
```

（2）利用牛顿插值求解根号 115 的近似值

```
% 编写牛顿插值法程序，求解
% 利用函数 f(x) = sqrt(x) 在 100，121，144 的值求 sqrt(115) 的近似值

% 函数定义
f = @(x) sqrt(x);

x = [100 121 144];
y = f(x);

% 牛顿插值法求解近似值
xx = 115;
yy = newton_interpolation(x, y, xx)
```

3.实验结果

```
>> exp2
yy =
    10.7228
```

可知牛顿插值法得到根号 115 的近似值为 10.7228。

装 订 线

实验 8：假定某天的气温变化记录如下，试用最小二乘方法找出这一天的气温变化规律：

表 4-13　实验 8 数据表

t/h	0	1	2	3	4	5	6	7	8	9	10	11	12
T/℃	15	14	14	14	14	15	16	18	20	22	23	25	28
t/h	13	14	15	16	17	18	19	20	21	22	23	24	
T/℃	31	32	31	29	27	25	24	22	20	18	17	16	

考虑下列函数类型，计算误差平方和，并作图比较效果。

（1）　二次函数

（2）　三次函数

（3）　四次函数

（4）　函数 C = aexp[-b(t - c)^2]

1.问题分析

不同阶数的多项式函数和指数函数在拟合数据点时,效果不同。

一般来说,随着函数的阶数或参数个数增加,函数表征能力也会增强,能更好地拟合给定的数据点,使误差平方和下降。但是阶数过高也会出现过拟合的问题,导致在训练数据外的泛化能力较低。

2.实验代码

（1）自定义多项式拟合函数

```
function p = nafit(x, y,m)
% 用途: 多项式拟合
% 格式: p = nafit(x, y, m)。x, y为数据向量, m为拟合多项式次数, p为返回多项式
% 系数降幂排列
A = zeros(m + 1, m + 1);
for i = 0 : m
    for j = 0 : m
        A(i + 1, j + 1) = sum(x.^(i + j));
    end
        b(i + 1) = sum(x.^i.*y);
end
a = A\b';
p = fliplr(a');
```

（2）试用最小二乘方法找出这一天的气温变化规律

```
% 试用最小二乘方法找出这一天的气温变化规律
```

```
% 考虑下列函数类型，计算误差平方和，并作图比较效果。
%  (1)    二次函数
%  (2)    三次函数
%  (3)    四次函数
%  (4)    函数C = aexp[-b(t - c)^2]

% 数据
hours = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24];
temperature = [15 14 14 14 14 15 16 18 20 22 23 25 28 31 32 31 29 27 25 24 22 20 18 17
16];

% 拟合二次函数
p2 = nafit(hours, temperature, 2);
y2 = polyval(p2, hours);
err2 = sum((y2 - temperature).^2);

% 拟合三次函数
p3 = nafit(hours, temperature, 3);
y3 = polyval(p3, hours);
err3 = sum((y3 - temperature).^2);

% 拟合四次函数
p4 = nafit(hours, temperature, 4);
y4 = polyval(p4, hours);
err4 = sum((y4 - temperature).^2);

% 拟合函数 C = aexp[-b(t - c)^2]
fun = @(p,x) p(1)*exp(-p(2)*(x-p(3)).^2);
p0 = [1,0.1,12]; % 初始参数
p = lsqcurvefit(fun,p0,hours,temperature);
y5 = fun(p, hours);
err5 = sum((y5 - temperature).^2);

% 绘图
figure;
plot(hours, temperature, 'ko', 'MarkerFaceColor', 'k'); hold on;
plot(hours, y2, 'r-', 'LineWidth', 2);
plot(hours, y3, 'g-', 'LineWidth', 2);
plot(hours, y4, 'b-', 'LineWidth', 2);
plot(hours, y5, 'm-', 'LineWidth', 2);
legend('气温', '二次函数', '三次函数', '四次函数', '高斯函数');
title('Least Squares Fit to Temperature Data');
hold off;

% 输出误差平方和
fprintf('二次函数误差平方和: %.2f\n', err2);
fprintf('三次函数误差平方和: %.2f\n', err3);
```

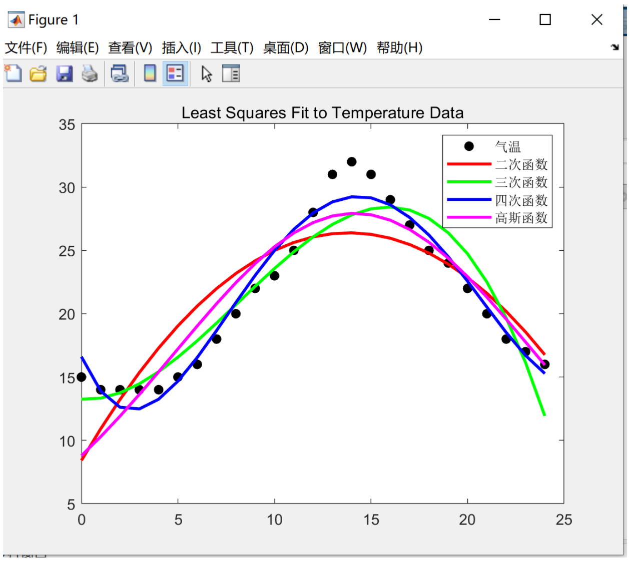
装

订

线

```
fprintf('四次函数误差平方和: %.2f\n', err4);
fprintf('高斯函数误差平方和: %.2f\n', err5);
```

3.实验结果



>> exp3

Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to

its initial value is less than the value of the function tolerance.

<stopping criteria details>

二次函数误差平方和: 241.24

三次函数误差平方和: 106.08

四次函数误差平方和: 36.28

高斯函数误差平方和: 144.79

五、实验总结

通过这次实验,我了解并掌握了拉格朗日插值法、牛顿插值法和最小二乘法在理论和实现上的不同之处。牛顿插值法收敛性高,但算法复杂;最小二乘法通过误差平方和最小化求解参数。

实验中我运用 MATLAB 实现了不同插值和拟合算法,观察了 Runge 现象和不同函数拟合效果。这帮助我深入理解相关概念和算法。

通过这次实验,我掌握了 MATLAB 基本函数的使用,能独立解决一定难度的数值问题。这为日后学习和工作打下基础。但在算法实现能力上还需努力。今后我将继续学习数值分析知识,运用更多工具进行科学计算。

装

订

线