**

**工程中的数值方法期中大作业**

|  |  |
| --- | --- |
| 学 院 | 机械与运载工程学院 |
| 授课教师 | 王琥 |
| 专业班级 | 车辆2004班 |
| 姓 名 | 庄淑晗 |
| 学 号 | 202004061131 |

2023年3月

# 目 录

一、计算机编程题……………………………………………………………………1

1、问题描述…………………………………………………………………………1

2、求解方法分析……………………………………………………………………1

2.1求解思路解析………………………………………………………………1

2.2拉格朗日插值法介绍………………………………………………………1

3、程序开发与结果分析……………………………………………………………2

3.1 程序流程图…………………………………………………………………2

3.2 程序说明……………………………………………………………………2

3.3 程序运行过程记录…………………………………………………………3

3.4 结果分析……………………………………………………………………3

3.5 程序原代码…………………………………………………………………8

二、思维发散题……………………………………………………………………10

**一、计算机编程题**

**1、问题描述**

基于函数f(x)=1/(1+x^2)，x∈[-5,5]，采用计算机编程的手段，分析验证拉格朗日插值法的龙格现象？

要求1：编程语言不限，但是需要提供源代码，以及软件运行过程和结果截图；

要求2：采用绘图函数，绘制插值函数曲线进行直观的对比。

**2、求解方法分析**

**2.1 求解思路解析**

运用重心拉格朗日法。它的优点是当插值点的个数增加一个时，将每个w_{j}都除以(x_{j}-x_{​{k+1}})，就可以得到新的重心权w_{​{k+1}}，计算复杂度为{\mathcal  O}(n)，比重新计算每个基本多项式所需要的复杂度{\mathcal  O}(n^{2})降了一个量级。

这种改进指的是速度上的，误差方面并没有什么差别，因此也会出现龙格现象。

**2.2 拉格朗日插值法介绍**

重心拉格朗日插值法是拉格朗日插值法的一种改进。在拉格朗日插值法中，运用多项式



可以将拉格朗日基本多项式重新写为：



定义重心权

w_{j}={\frac  {1}{\prod _{​{i=0,i\neq j}}^{k}(x_{j}-x_{i})}}

上面的表达式可以简化为：

\ell _{j}(x)=\ell (x){\frac  {w_{j}}{x-x_{j}}}

于是拉格朗日插值多项式变为：



即所谓的重心拉格朗日插值公式（第一型）或改进拉格朗日插值公式。它的优点是当插值点的个数增加一个时，将每个w_{j}都除以(x_{j}-x_{​{k+1}})，就可以得到新的重心权w_{​{k+1}}，计算复杂度为{\mathcal  O}(n)，比重新计算每个基本多项式所需要的复杂度{\mathcal  O}(n^{2})降了一个量级。

将以上的拉格朗日插值多项式用来对函数g(x)\equiv 1插值，可以得到：

\forall x,\,g(x)=\ell (x)\sum _{​{j=0}}^{k}{\frac  {w_{j}}{x-x_{j}}}

因为g(x)\equiv 1是一个多项式。

因此，将L(x)除以g(x)后可得到：

L(x)={\frac  {\sum _{​{j=0}}^{k}{\frac  {w_{j}}{x-x_{j}}}y_{j}}{\sum _{​{j=0}}^{k}{\frac  {w_{j}}{x-x_{j}}}}}\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ (2)

这个公式被称为重心拉格朗日插值公式（第二型）或真正的重心拉格朗日插值公式。它继承了（1）式容易计算的特点，并且在代入*x*值计算L(x)的时候不必计算多项式\ell (x)。

**3、程序开发与结果分析**

**3.1 程序流程图**

|  |  |  |
| --- | --- | --- |
| 原函数f(x)=1/(1+x^2) | 运用重心拉格朗日插值公式 | Plot()函数绘制图像 |

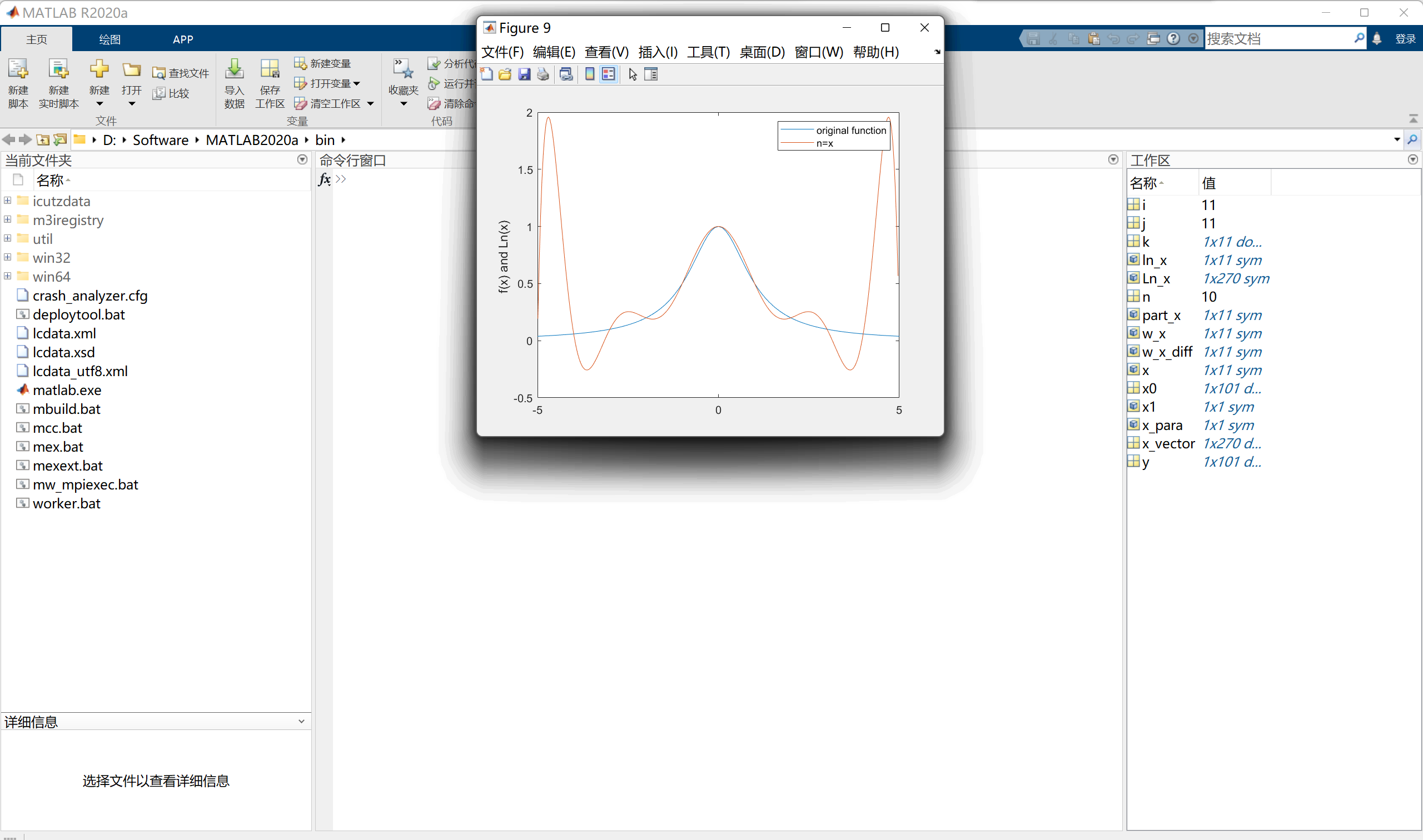
**3.2 程序说明**

采用Matlab语言

x0、y为被插值函数变量,n为等分数，x(k)为插值点,W\_x为Li(x)的分子乘（x-x(j),W1\_x为Li(x)的分母,Ln\_x为插值函数多项式

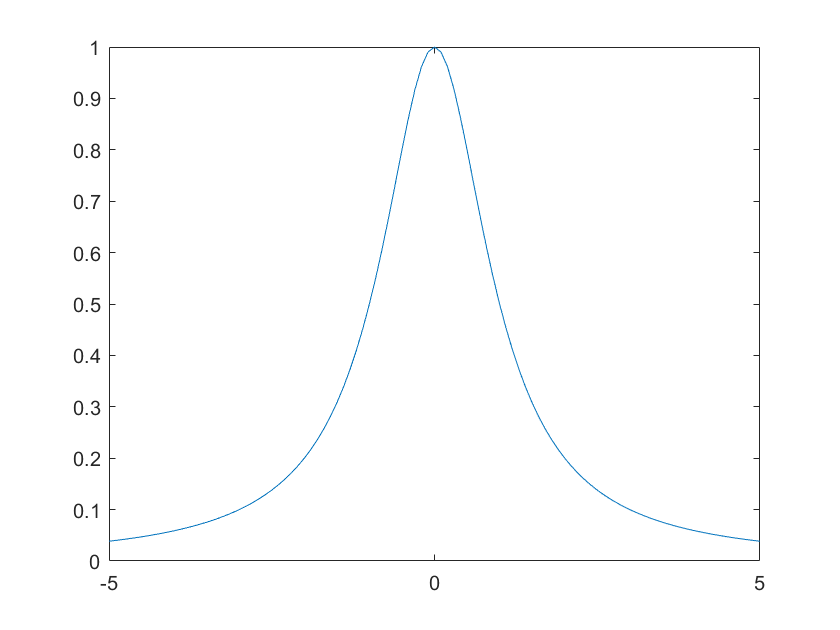
拟合由真实分布产生的点，得到的多项式函数并不是次数越高就和真实函数越接近。相反，本例中次数越高，结果偏离越大。我们需要做一些先验工作，了解真实函数的曲线走向，再对症下药。即在不熟悉曲线运动趋势的前提下，不要轻易使用高次多项式。

**3.3 程序运行过程记录**

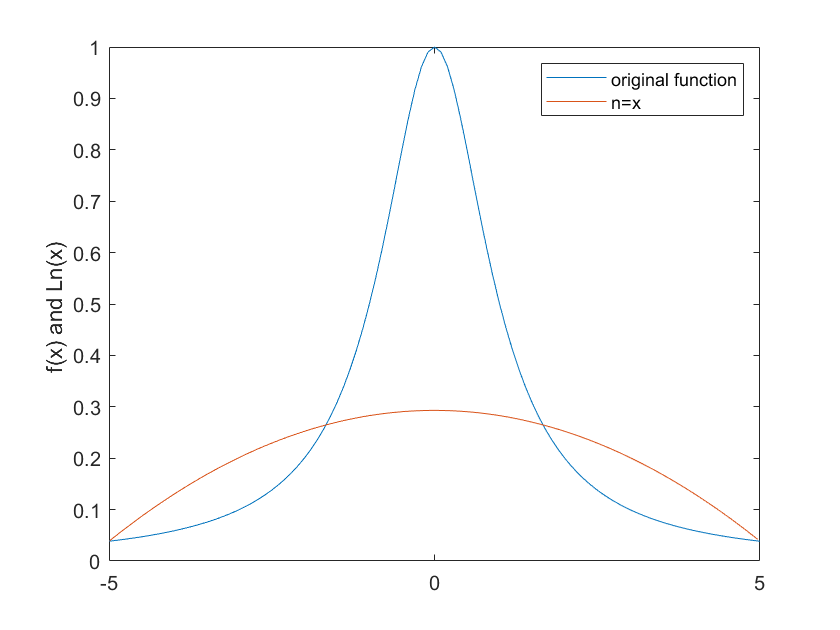
****

3.3图：程序运行过程

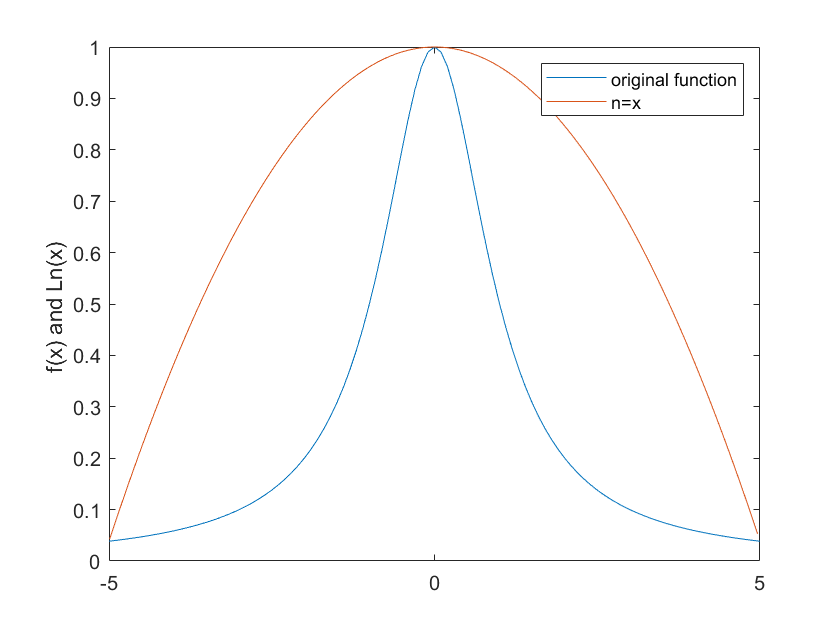
**3.4 结果分析**



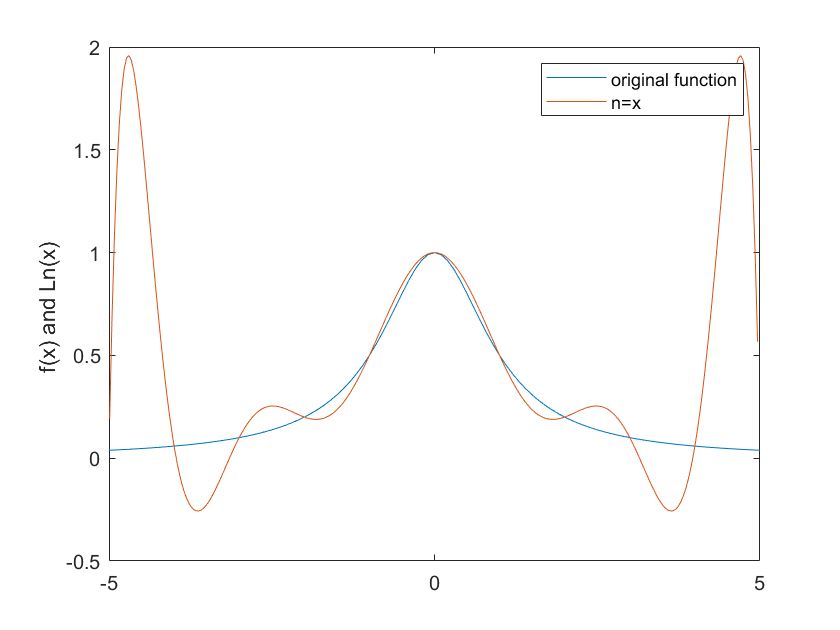
3.4图1：原函数



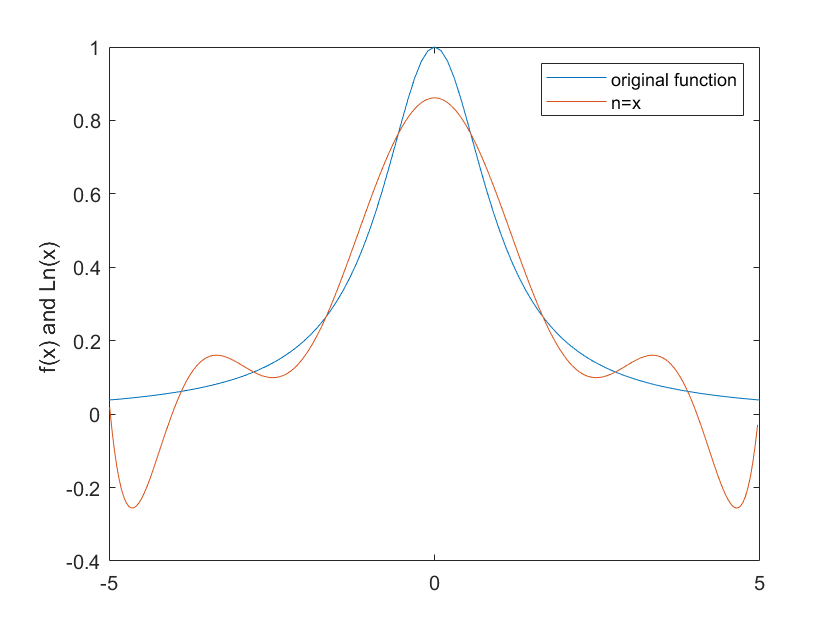
3.4图2：n=2



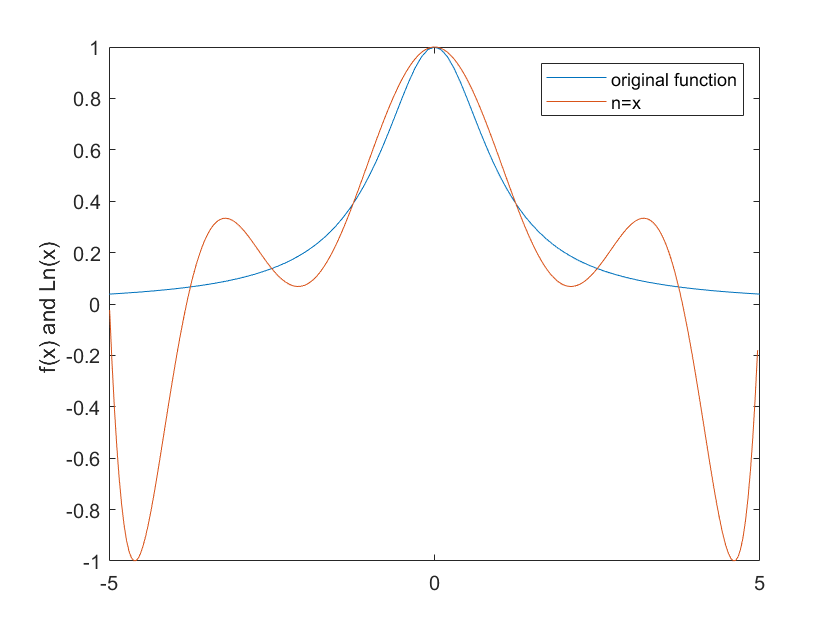
3.4图3：n=3



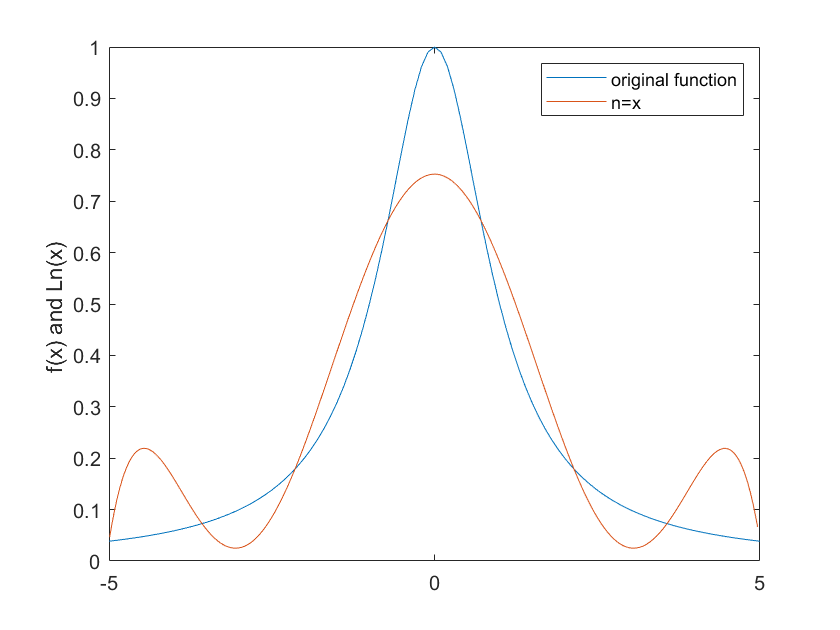
3.4图4：n=4



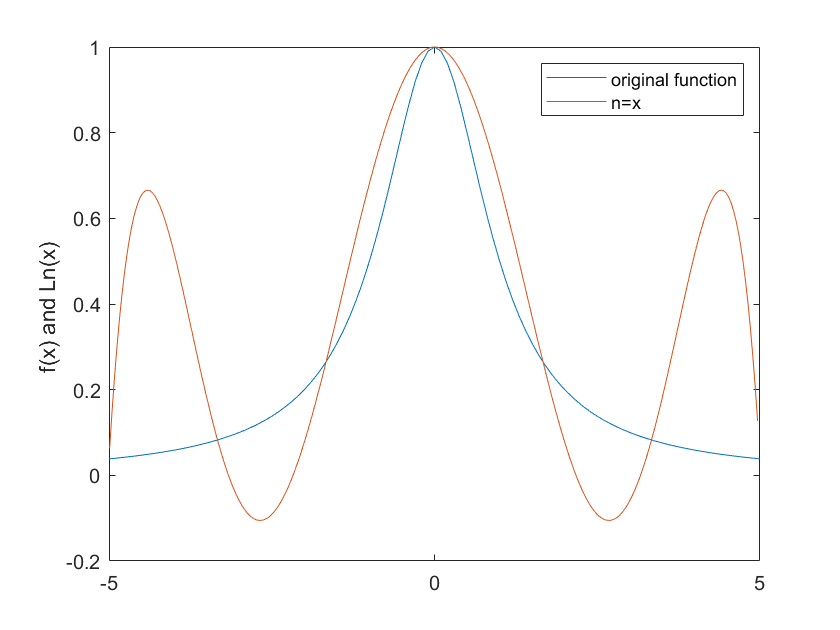
3.4图5：n=5



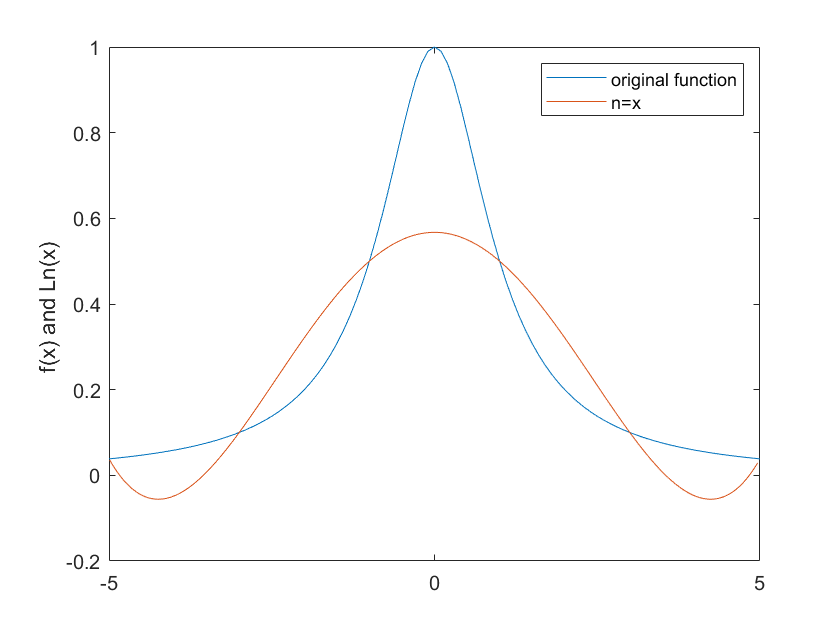
3.4图6：n=6

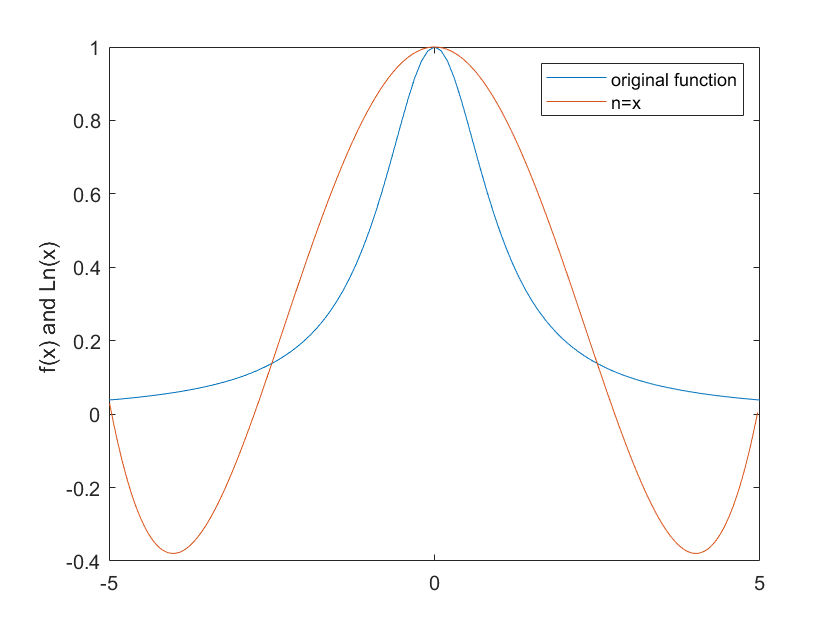


3.4图7：n=7



3.4图8：n=8



3.4图9：n=9

3.4图10：n=10

**3.5 程序原代码**

clc

clear

syms x0 x1 y x n x\_para

x0=-5:0.1:5;

y=1./(1+x0.^2);

for n=2:10

k=0:n; %k的范围从0到n,共n+1项

x(k+1)=-5+10\*k/n; %从x（1）到x（n+1），共n+1项，即从第0点到第n点

for i=1:(n+1) %共k+1项

part\_x(i)=(x\_para-x(i)); %每个因式的组成，共k+1种

end

w\_x(n+1)=prod(part\_x); %求w（x）

w\_x\_diff(n+1)=diff(w\_x(n+1));

for j=1:(n+1)

w\_x\_diff(j)=subs(w\_x\_diff(n+1),x\_para,x(j));

ln\_x(j)=(1/(1+x(j)^2)) \* w\_x(n+1) / ( (x\_para-x(j))\*(w\_x\_diff(j)) );

end

Ln\_x=sum(ln\_x);

x\_vector=-4.99:0.037:4.99;

Ln\_x=subs(Ln\_x,x\_para,x\_vector);

figure

plot(x0,y)

hold on

plot(x\_vector,Ln\_x)

legend ('original function','n=x')

ylabel('f(x) and Ln(x) ');

end

**二、思维发散题**

根据表1 ：2022年长沙各月份平均空气质量指数分析长沙空气质量的特点。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1月** | **2月** | **3月** | **4月** | **5月** | **6月** |
| 152 | 70 | 78 | 47 | 59 | 51 |
| **7月** | **8月** | **9月** | **10月** | **11月** | **12月** |
| 52 | 38 | 63 | 78 | 79 | 93 |

表1 ：2022年长沙各月份平均空气质量指数

clear

clc

%% 原始散点

t = [1 : 12];

y = [152 70 78 47 59 51 52 38 63 78 79 93];

plot(t, y, 'o');

%% 二次多项式拟合

p = polyfit(t, y, 2);

hold on

plot(t, polyval(p, t), 'r'); % 画多项式图像

%% 三次多项式拟合

p = polyfit(t, y, 3);

hold on

plot(t, polyval(p, t), 'k'); % 画多项式图像

%% 四次多项式拟合

p = polyfit(t, y, 4);

hold on

plot(t, polyval(p, t), 'm'); % 画多项式图像

%% 十次多项式拟合

p = polyfit(t, y, 10);

hold on

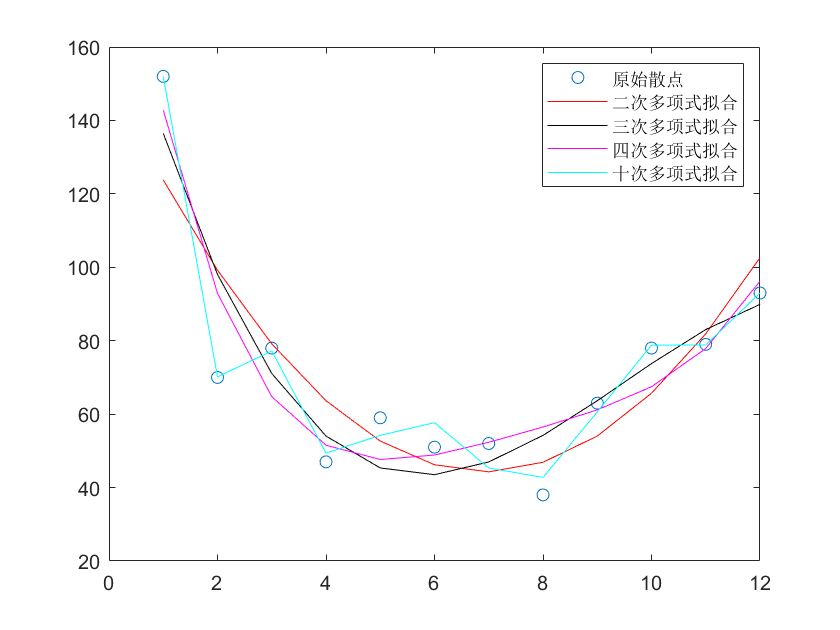
plot(t, polyval(p, t), 'c'); % 画多项式图像

%% 图例

legend('原始散点', '二次多项式拟合', '三次多项式拟合', ...

'四次多项式拟合', '十次多项式拟合')

IMG_256

IMG_256