

《数值分析》上机综合大作业成绩

上机实验报告书

**2021 / 2022 学年 第 2 学期**

**课程名称： 数值分析**

**专业班级： 计算机2001**

**学 号： 200405129**

**姓 名： 刘冉**

**指导教师： 李威**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 评分项目 | | 分值比例 | 实际得分 |
| 1 | 出勤情况 | | 10% |  |
| 2 | 技术选择合理性 | | 10% |  |
| 3 | 编程质量和答辩 | 设计方案合理、正确 | 10% |  |
| 对设计过程进行阐述 | 10% |  |
| 程序结果演示 | 10% |  |
| 回答问题 | 10% |  |
| 4 | 设计报告 | 文字描述和说明 | 10% |  |
| 图表清楚正确 | 10% |  |
| 成果体现 | 10% |  |
| 撰写规范性 | 10% |  |
| 总得分/成绩等级 | | | |  |
| 备注： | | | | |

**题目1 多项式插值算法**

**1．1 题目的主要研究内容及预期达到的目标（宋体四号加粗左对齐）**

1. **研究内容**

设 f(x)=1/(1+x2) x∈[-5,5]

在[-5,5]内取n+1个等距节点xk=-5+10/n\*k (k=0,1,2,L ,n)，构造当 n=2,4,6,8,10时的插值多项式Ln(x),并在同一张图上画出f(x)和所有Ln(x)的图形。

题目主要研究多项式插值算法，根据插值节点的数量做n=2,4,6,8,10时，各阶插值多项式的图像，并与原函数图像进行对比。

1. **预期目标**

利用拉格朗日插值法做出个阶插值多项式图像，并与原函数进行对比

找出阶数增加时，插值函数变化的趋势。

**1．2 题目研究的工作基础或实验条件**

1. **硬件环境**

本程序的硬件操作环境如表1所示。

　　　　 表 1 硬件环境

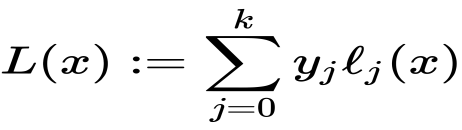
|  |  |
| --- | --- |
| 操作系统 | Windows 10 64Bit |
| CPU | Intel(R) Core(TM) i5-1035G1 |
| 显卡 | NVIDIA　GeForce　MX230 |
| 内存 | 8.00G |

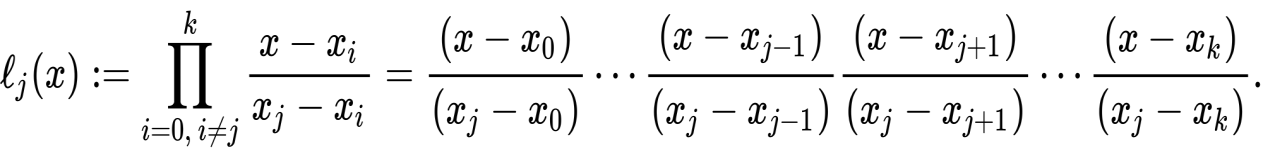
1. **软件环境**

　　本程序使用的编程语言为Python，作图时使用Python的matplotlib库，软件使用Visual stdio 2022

**1．3 设计思想**

拉格朗日插值多项式为：

{\displaystyle L(x):=\sum \_{j=0}^{k}y\_{j}\ell \_{j}(x)}

{\displaystyle \ell \_{j}(x):=\prod \_{i=0,\,i\neq j}^{k}{\frac {x-x\_{i}}{x\_{j}-x\_{i}}}={\frac {(x-x\_{0})}{(x\_{j}-x\_{0})}}\cdots {\frac {(x-x\_{j-1})}{(x\_{j}-x\_{j-1})}}{\frac {(x-x\_{j+1})}{(x\_{j}-x\_{j+1})}}\cdots {\frac {(x-x\_{k})}{(x\_{j}-x\_{k})}}.}

设计思想如下：

1. 根据n值计算出n+1个插值节点的坐标
2. 根据插值节点的坐标可得对应n值下的拉格朗日多项式
3. 利用for循环画出n取不同值时的插值函数图像以及原函数图像

**1．4 流程图**

程序流程图如图1所示

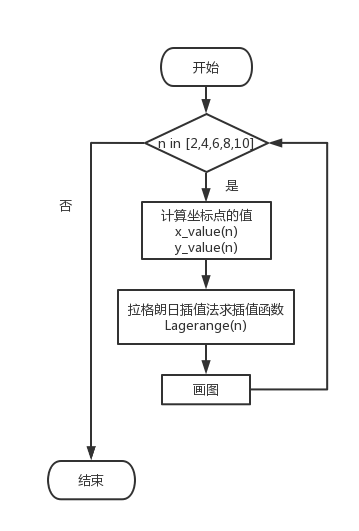
****

图1 流程图

拉格朗日插值法求插值函数算法流程图如图2

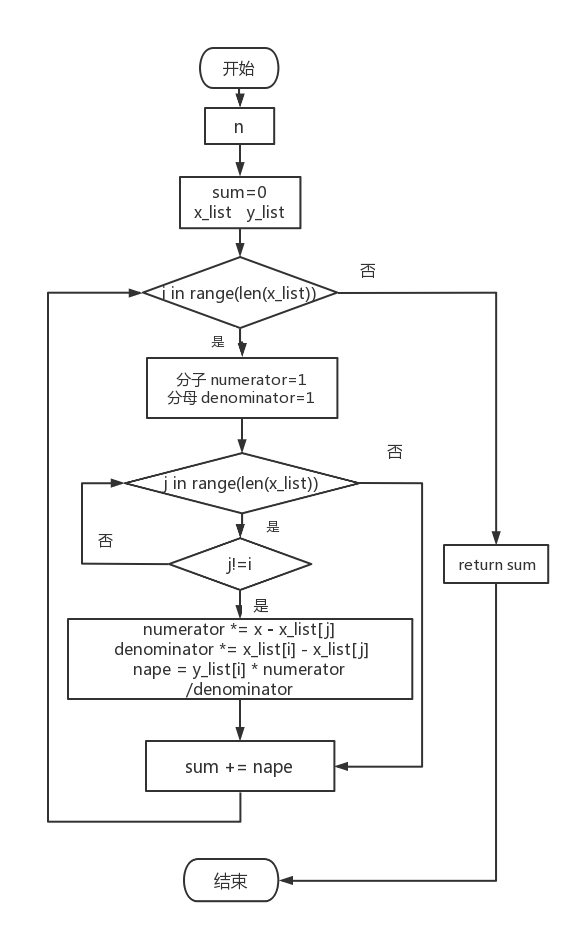
****

图2拉格朗日插值法求插值函数算法流程图

**1．5 主要程序代码(要求必须有注释)**

import matplotlib.pyplot as plt

import numpy as np

#1.计算插值节点x的值

def x\_value(n):

x\_point=[]

k=0

while k<=n:

t=-5+10\*k/n

x\_point.append(t)

k+=1

return x\_point

#2.计算插值节点y的值

def y\_value(X\_point):

y\_point=[]

for ch in X\_point:

m=1/(1+ch\*ch)

y\_point.append(m)

return y\_point

#3.拉格朗日插值函数

def Lagrange(n):

sum= 0

x\_list=x\_value(n)

y\_list=y\_value(x\_list)

for i in range(len(x\_list)):

numerator = 1

denominator = 1

for j in range(len(x\_list)):

if (j != i):

numerator \*= x - x\_list[j] #每一项中的分子

denominator \*= x\_list[i] - x\_list[j] #每一项中的分母

nape = y\_list[i] \* numerator / denominator #每一项

sum += nape

return sum

#4.创建figure窗口

plt.figure(num=3, figsize=(8, 5))

#5.标题设置

plt.xlabel("X") #x轴标题

plt.ylabel("Y") #y轴标题

#6，轴设置

plt.xlim((-6,6)) #设置x轴范围

plt.ylim((-1,2)) #设置y轴范围

#7.画图

x = np.arange(-5, 5, 0.01)

for n in[2,4,6,8,10]:

plt.plot(x, Lagrange(n),label = 'n='+str(n))

plt.plot(x,y\_value(x),label = 'n=orginal')

plt.legend(loc='lower center')

plt.savefig('Lagrange,png')

plt.show()

**1．6 运行结果及分析**

运行结果如图2

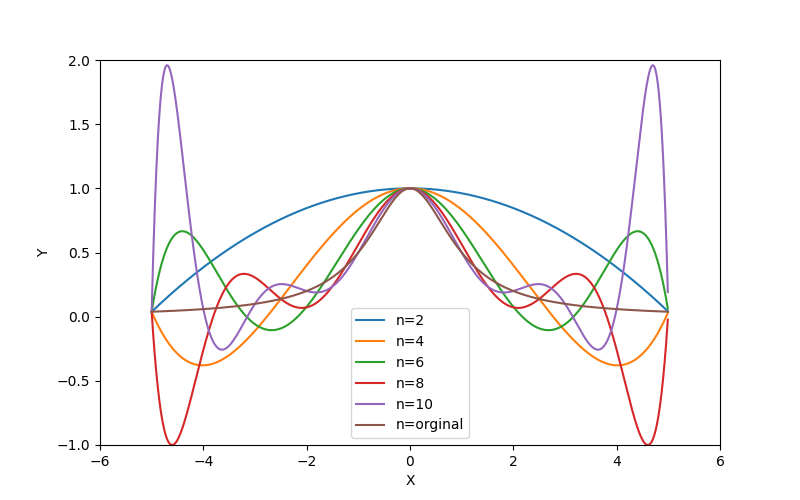


图 2 运行结果

从图中可以看出，n取2，4，，6，810时，插值的阶数越高，插值函数在插值区间中部与原函数拟合的越好，但在插值阶数较高的时候，虽然插值函数过没一个插值节点，但误差极大。所以，选择适当的插值阶数尤为重要

**1．7 心得体会**

数值分析方面：通过本次上机实验，我更好的理解了拉格朗日插值算法，直观的感受了利用拉格朗日插值求解出的插值函数所存在的龙格现象。

编程方面：1.利用matplotlib画图时比较困惑在for循环里改变颜色标签，查找资料得知for循环换下将自动循环显示默认颜色 2.如何设置自变量x，绘制出y=f(x)的线条，查找得知设置x = np.arange(-5, 5, 0.01)，**返回**一个5为终点，-5起点的固定步长为0.01的排列，即可绘图。