

《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2002班\_\_\_\_\_\_\_**

**学 号：\_\_\_\_\_\_\_\_\_200405220\_\_\_\_\_\_\_\_**

**姓 名：\_\_\_\_\_\_\_\_\_\_周蓬睿\_\_\_\_\_\_\_\_\_\_**

**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 插值函数运算**

**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 IMG_256 x属于[-5,5]，在[-5,5]内取n+1个等距节点 IMG_257 构造当n=2,4,6,8,10时的拉格朗日插值多项式 IMG_258 ,并在同一张图上画出f(x)和所有 IMG_258 的图形；

（2）预期目标：通过编程求出各个拉格朗日插值并画出原函数及各个插值函数的图象。

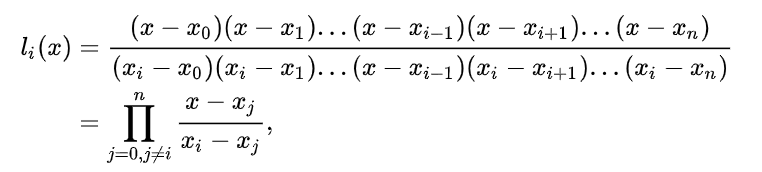
**1．2 题目研究的工作基础或实验条件**

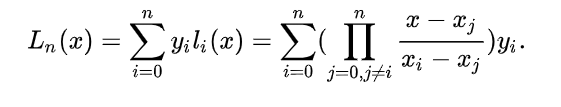
（1）硬件环境：笔记本 Acer Nitro AN515-55；

（2）软件环境：Window 10家庭中文版、Visual Studio 2019、EasyX图形库。

**1．3 设计思想**

设计思想为拉格朗日插值法思想，过程如下：

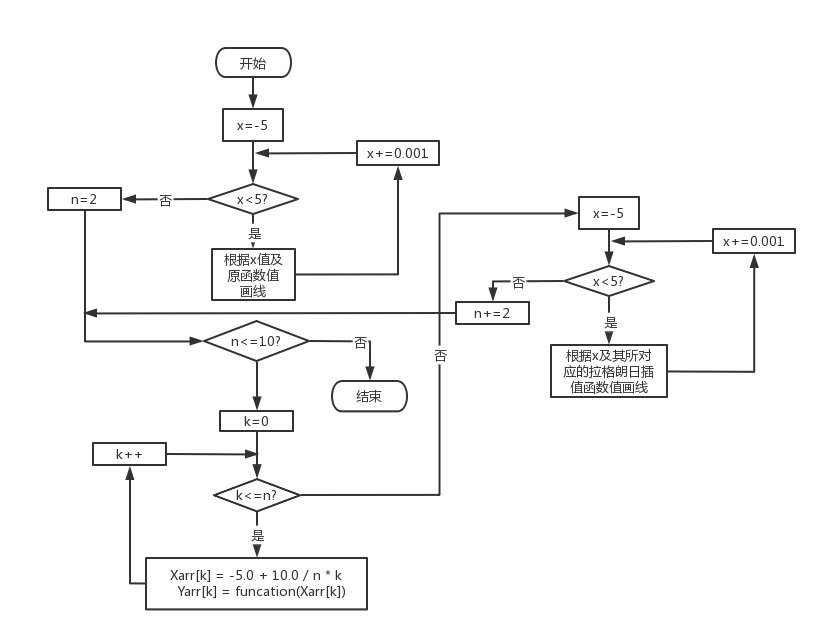


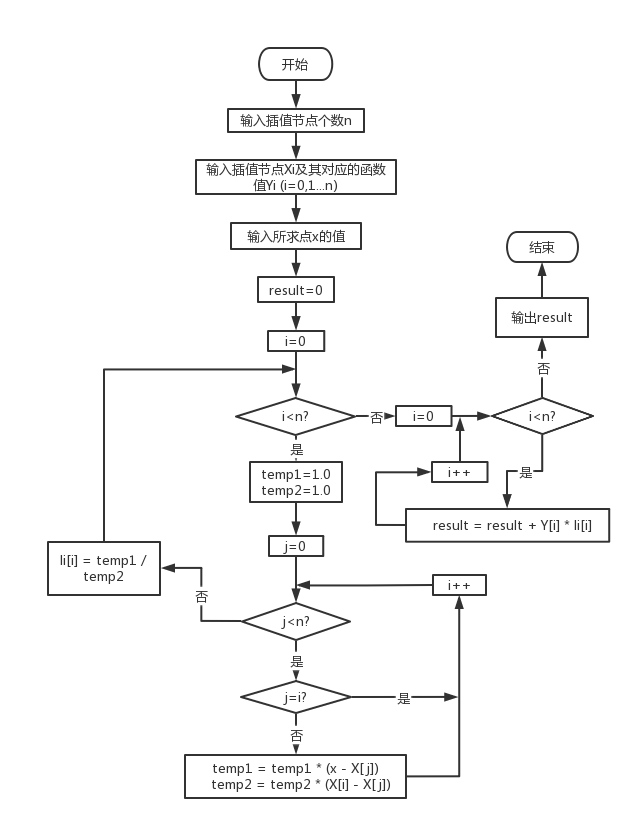


其中x0到xn是处于插值区间的各个递增的插值节点，y0到yn是各个插值节点所对应的函数值。

**1．4 流程图**

绘制函数图象的流程图如图一所示，通过拉格朗日插值法计算x对应的插值函数值的流程图如图二所示。



图一 总体流程图****

图二 局部流程图

**1．5 主要程序代码(要求必须有注释)**

float Lagrange(float X[], float Y[], int n, float x)//采用Lagrange插值方法，分别表示，长度，深度，数组数目和所需要计算点的值

{

float result = 0.0;//最后得到的函数解

float li[200];//lagrange基本插值多项式

float temp1, temp2;//分别用来存储Lagrange多项式的分母和分子

int i, j;//循环变量

for (i = 0; i < n; i++){//Lagrange插值算法

temp1 = 1.0; temp2 = 1.0;

for (j = 0; j < n; j++){

if (i == j)

continue;

temp1 = temp1 \* (x - X[j]);

temp2 = temp2 \* (X[i] - X[j]);

}

li[i] = temp1 / temp2;

}

for (i = 0; i < n; i++){

result = result + Y[i] \* li[i];

}

return result;

}

int main()

{

//两个存储所选节点的数组

float\* Xarr;

float\* Yarr;

initWindow();//打开画板

for (float x = -5; x < 5; x += step) {//画原函数图像

drawLine(x, funcation(x), x+step, funcation(x+step));

}

for (int n = 2; n <= 10; n += 2) {//画插值函数图像

Xarr = new float[n + 1];

Yarr = new float[n + 1];

for (int k = 0; k <= n; k++) {//循环生成不同的n所对应的节点组

Xarr[k] = -5.0 + 10.0 / n \* k;

Yarr[k] = funcation(Xarr[k]);

}

for (float x = -5; x < 5; x += step) {//画不同插值函数图

drawLine(x, Lagrange(Xarr, Yarr, n + 1, x), x + step, Lagrange(Xarr, Yarr, n + 1, x + step));

}

delete[]Xarr;

delete[]Yarr;

}

\_getch();

closegraph();

return 0;

}

**1．6 运行结果及分析**

运行结果如图三所示。

图三 运行结果

图三中黑色曲线为原函数曲线，红色曲线为n=2时的曲线，黄色曲线为n=4时的曲线，绿色曲线为n=6时的曲线，蓝色曲线为n=8时的曲线，棕色曲线为n=10时的曲线。

从图三中可以看出，当n分别等于2、4、6、8、10时，随着n的增大 ,插值函数在插值区间中部与原函数的拟合效果越好，但在区间两端出现了龙格现象。各个曲线与原函数的交点个数与在插值区间内所选节点个数相同。因此，当插值阶数较高时，选用拉格朗日插值法会使得结果误差极大。

**1．7 心得体会**

在实验过程中进行模拟画图时出现了问题，由于对这方面不够熟悉，相关经验不足，导致无从下手，经过一番思考，使用选取相隔极近两点连接直线的方法完成模拟画图，问题得以解决。就实验结果来说，当插值阶数较大时，不能在选用拉格朗日插值法计算结果，应当选用更加合适的方法来计算。