

**学 生 实 验 报 告**

**（理工类）**



课程名称： 计算方法 专业班级： 15软件工程（Z）

学生学号： 1512001066 学生姓名： 吴跟强

所属院部： 软件工程学院 指导教师： 谢维奇

**20 17 ——20 18 学年 第 1 学期**

金陵科技学院教务处制

**实验报告书写要求**

实验报告原则上要求学生手写，要求书写工整。若因课程特点需打印的，要遵照以下字体、字号、间距等的具体要求。纸张一律采用A4的纸张。

**实验报告书写说明**

实验报告中一至四项内容为必填项，包括实验目的和要求；实验仪器和设备；实验内容与过程；实验结果与分析。各院部可根据学科特点和实验具体要求增加项目。

**填写注意事项**

（1）细致观察，及时、准确、如实记录。

（2）准确说明，层次清晰。

（3）尽量采用专用术语来说明事物。

（4）外文、符号、公式要准确，应使用统一规定的名词和符号。

（5）应独立完成实验报告的书写，严禁抄袭、复印，一经发现，以零分论处。

**实验报告批改说明**

实验报告的批改要及时、认真、仔细，一律用红色笔批改。实验报告的批改成绩采用百分制，具体评分标准由各院部自行制定。

**实验报告装订要求**

实验批改完毕后，任课老师将每门课程的每个实验项目的实验报告以自然班为单位、按学号升序排列，装订成册，并附上一份该门课程的实验大纲。

实验项目名称： 方程求根 实验学时： 2

同组学生姓名： 实验地点： 1514

实验日期： 11.28 实验成绩：

批改教师： 批改时间：

一、实验目的和要求

实验目的：

1. 通过对二分法、牛顿法、割线法作编程练习，进一步体会它们各自不同的特点；
2. 了解二分法，切线法，割线法。
3. 能熟练运用二分法，牛顿法进行方程求根
4. 通过上机调试运行，对方程求根的几种方法程序进行改进。

实验要求：

1. 上机前作好充分准备，包括复习编程所需要的语言工具。
2. 上机时要遵守实验室的规章制度，爱护实验设备。
3. 记录调试过程及结果，记录并比较与手工运算结果的异同。
4. 程序调试完后，须由实验辅导教师在机器上检查运行结果。
5. 给出本章实验单元的实验报告。

二、实验仪器和设备

1. 硬件设备：IBM PC以上计算机，有硬盘和一个软驱、单机和网络环境均可。
2. 软件环境： C语言运行环境。

三、实验原理、方法

**二分算法计算步骤：**

（1）输入有根区间的端点a、b及预先给定的精度ε；

（2）计算中点x=(a+b)/2；

（3）若f(x)f(b)<0，则a=x，转向下一步；否则b=x，转向下一步；

（4）若b-a<ε，则输出方程满足精度要求的根x，结束；否则转向步骤（2）。

**牛顿迭代法：**

牛顿迭代法是一种逐步线性化方法，即将非线性方程f(x)=0的求根问题归结为计算一系列线性方程的根。

设xk是方程f(x)=0的一个近似根，将f(x)在xk处作一阶泰勒展开，即

f(x)≈f(xk)+f′(xk)(x- xk)

于是得到如下的近似方程

f(xk)+f′(xk)(x- xk)=0 （2.7）

设f′(xk)≠0，则式（2.7）的解为



取x作为原方程的新的近似根xk+1，即令

　　　k=0,1,2, … 　　 （2.8）

则称式（2.8）为牛顿迭代公式。用牛顿迭代公式（2.8）求方程近似根的方法称为牛顿迭代法，简称牛顿法，又称切线法。

四、实验内容

1. 以方程：x3-0.2x2-0.2x-1.2=0为例，编写程序求方程的根
2. 编写二分法、迭代法、牛顿法程序，分析运行结果。
3. 对用这两种方法求解出的根进行对比分析

五、实验过程

二分法程序：

#include "stdio.h"

#include "math.h"

double f(double x)

{

double sum;

sum=x\*x\*x-0.2\*x\*x-0.2\*x-1.2;

return sum;

}

double judge(double a,double b,double c)

{

double x;

x=(a+b)/2;

while(fabs(b-a)>=c)

{

double flag=f(x)\*f(b);

if(flag<0)

a=x;

else

b=x;

x=(a+b)/2;

}

return x;

}

int main()

{

double a,b,c,x=0.0;

printf("请输入有根区间a、b及预先给定的精度c：");

scanf("%lf,%lf,%lf",&a,&b,&c);

x=judge(a,b,c);

printf("满足精度要求的根x=%lf",x);

return 0;

}

牛顿迭代法程序：

#include "stdio.h"

#include "math.h"

double f(double x)

{

double sum;

sum=x\*x\*x-0.2\*x\*x-0.2\*x-1.2;

return sum;

}

double f1(double x)

{

double sum1;

sum1=3\*x\*x-0.4\*x-0.2;

return sum1;

}

int main()

{

double x0,c,x1;

int N=0,i=1;

printf("请输入x0、N及预先给定的精度c：");

scanf("%lf,%d,%lf",&x0,&N,&c);

while(f1(x0)!=0&&fabs(x1-x0)>c&&i<N)

{

x0=x1;

x1=x0-f(x0)/f1(x0);

i++;

}

if(fabs(x1-x0)<=c)

printf("满足精度要求的根x=%lf",x1);

if(i==N)

printf("不符合要求");

if(f1(x0)==0)

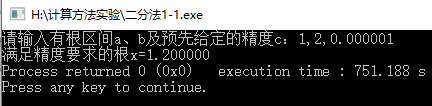
printf("x=%lf",x0);

return 0;

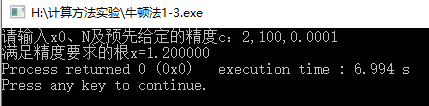
}

六、实验结果与分析

二分法程序结果：



牛顿迭代法程序结果：



对用这两种方法求解出的根进行对比分析

答：结果都是1.2这个根，根据运行时间来看，牛顿迭代法比二分法时间要节省很多，效率更高。

七、实验感想

根据这次实验，对二分法和牛顿迭代法更加的清楚了，对方法的运用更加熟悉，也知道了方法的使用会影响结果的精确度，牛顿法比二分法考虑的要全面些。

实验项目名称： 线性方程组数值解法实验学时： 2

同组学生姓名： 实验地点： 1514

实验日期： 12.5 实验成绩：

批改教师： 批改时间：

一、实验目的

1．掌握方程组的解法，迭代法及其收敛性。

2．能熟练掌握高斯消去法，列主元高斯消去法，三角分解法。

3．掌握雅可比迭代法，高斯=赛德尔迭代求线性方程组的解。

二、实验要求

1．上机前作好充分准备，比较不用的方法解决相同问题的不同。

2．上机时要遵守实验室的规章制度，爱护实验设备。

3．记录调试过程及结果，记录并比较与手工运算结果的异同。

4．程序调试完后，须由实验辅导教师在机器上检查运行结果。

5．给出本章实验单元的实验报告。

三、实验设备、环境

1．硬件设备：IBM PC以上计算机，有硬盘和一个软驱、单机和网络环境均可。

2．软件环境： C语言运行环境。

四、实验内容

1．求解方程组：

（1）

（2）

2．编写高斯消去法、三角分解法程序，分析运行结果。

3．调试运行列主元高斯消去法、列主元三角分解法算法程序。

4．并用上述几种算法程序计算出上面两个方程组的解。

1. **实验代码**

**高斯消去法：**

#include <stdio.h>

#include <math.h>

#define Max 10

void ColPivot(float a[Max][Max],float b[],int n)//列主元高斯

{

int i,j,k,m\_i;

float m\_x,temp;

for(i=0;i<n-1;i++)

{

j=i+1; m\_i=i; m\_x=fabs(a[i][i]);

for(;j<n;j++)

if(fabs(a[j][i]>m\_x)) /\*找主元素\*/

{

m\_i=j;

m\_x=fabs(a[j][i]);

}

if(i<m\_i) /\*交换两行\*/

{

temp=b[i];

b[i]=b[m\_i];

b[m\_i]=temp;

for(j=i;j<n;j++)

{

temp=a[i][j];

a[i][j]=a[m\_i][j];

a[m\_i][j]=temp;

}

}

/\*消元\*/

for(j=i+1;j<n;j++)

{

temp=-a[j][i]/a[i][i];

b[j]+=b[i]\*temp;

for(k=i;k<n;k++)

a[j][k]+=a[i][k]\*temp;

}

}

}

int main()

{

int i,j,k,n;

float a[Max][Max],b[Max],x[Max];

printf("请输入矩阵维数n：");

scanf("%d",&n);

if(n>Max||n<=0)

return 1;

printf("Input the A(i,j):\n");

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

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

scanf("%f",&a[i][j]);

printf("Input b(i):\n");

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

scanf("%f",&b[i]);

ColPivot(a,b,n);

x[n-1]=b[n-1]/a[n-1][n-1]; /\*解方程\*/

for(i=n-2;i>=0;i--)

{

x[i]=b[i];

for(j=i+1;j<n;j++)

x[i]-=a[i][j]\*x[j];

x[i]/=a[i][i];

}

printf("方程组的解:\n");

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

printf("x[%d]=%f\n",i,x[i]);

return 0;

}

**三角分解法：**

#include <stdio.h>

#include <math.h>

#define Max 10

void \*LU(float a[Max][Max],int n,float b[])//进行LU分解

{

int i,j,k;

float y[Max],L[Max][Max],U[Max][Max],x[Max];

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

U[i][i]=1;/\*U矩阵对角元素赋值为1\*/

for(k=0;k<n;k++)

{

for(i=k;i<n;i++) /\*计算L矩阵的第k列元素\*/

{

L[i][k]=a[i][k];

for(j=0;j<k;j++)

L[i][k]-=(L[i][j]\*U[j][k]);

}

for(j=k;j<n;j++) /\*计算U矩阵的第k行元素\*/

{

U[k][j]=a[k][j];

for(i=0;i<k;i++)

U[k][j]-=(L[k][i]\*U[i][j]);

U[k][j]/=L[k][k];

}

}

for(i=0;i<n;i++) /\*计算Ly=b中的y\*/

{

y[i]=b[i];

for(j=0;j<i;j++)

y[i]-=(L[i][j]\*y[j]);

y[i]/=L[i][i];

}

for(i=n-1;i>=0;i--) /\*计算Ux=y中的x\*/

{

x[i]=y[i];

for(j=n-1;j>=i+1;j--)

x[i]-=(U[i][j]\*x[j]);

x[i]/=U[i][i];

}

printf("方程组的解:\n");

for(k=0;k<n;k++)

printf("x[%d]=%f\n",k,x[k]);

}

int main()

{

int n;

int i,j,k;

float a[Max][Max],b[Max],\*x;

printf("输入线性方程组的维数:");

scanf("%d",&n);

if(n>Max||n<=0)

return 1;//错误处理，维度n要小于Max

printf("Input the A(i,j):\n");

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

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

scanf("%f",&a[i][j]);

printf("Input b(i):\n");

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

scanf("%f",&b[i]);

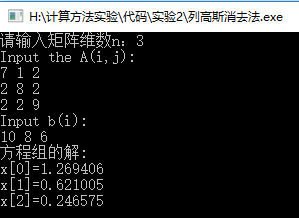
LU(a,n,b);

return 0;

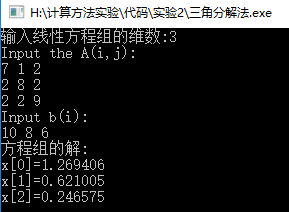
}

1. **实验结果与分析**
2. 

高斯消去法：

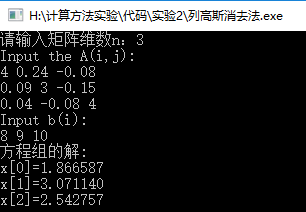


三角分解法：

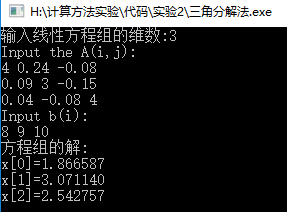


（2）

高斯消去法：



三角分解法：



**七、实验心得**

这次试验我更加了解了高斯消去法和三角分解法的用法，更加熟练运用这两种方法了。这个三角分解的算法一直有点问题，，造成结果与高斯方法不同。后面改进了一下算法。方法并不是想想就可以解决的，还需要步步完善！过程中也逐渐对这两种方法熟悉了解了。

实验项目名称： 插值法 实验学时： 2

同组学生姓名： 实验地点： 1514

实验日期： 12.12 实验成绩：

批改教师： 批改时间：

一、实验目的

1．掌握插值函数的概念，插值多项式的唯一性。

2．掌握插值余项，差分及等距插值公式，高次插值的误差分析。

3．掌握基本插值多项式，拉格朗日插值多项式，差商，牛顿插值多项式。

二、实验要求

1．上机前作好充分准备，比较不用的方法解决相同问题的不同。

2．上机时要遵守实验室的规章制度，爱护实验设备。

3．记录调试过程及结果，记录并比较与手工运算结果的异同。

4．程序调试完后，须由实验辅导教师在机器上检查运行结果。

5．给出本章实验单元的实验报告。

三、实验设备、环境

1．硬件设备：IBM PC以上计算机，有硬盘和一个软驱、单机和网络环境均可。

2．软件环境： C语言运行环境。

四、实验内容

1．已知函数表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 1.615 | 1.634 | 1.702 | 1.828 | 1.921 |
| Y=f(x) | 2.41450 | 2.46459 | 2.65271 | 3.03035 | 3.34066 |

（1）用拉格朗日插值（二次、四次）计算f(1.682)和f(1.813)的近似值。

（2）构造出均差表，并利用牛顿（均差）插值多项式计算f(1.682)和f(1.813)的近似值。

（3）分析并比较两种算法得到的近似值的精度。

2．编写拉格朗日和牛顿插值算法程序，分析运行结果。

**五、实验代码**

**拉格朗日插值代码：**

#include "stdio.h"

#define N 5//xi的个数

double Lagrange(double x[],double y[],double xx,int n)

//xx:x点的x值，n：表示n次拉格朗日插值

{

int i,j;

double f[N],yy=0;

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

{

f[i]=y[i];

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

if(j!=i)

f[i]\*=(xx-x[j])/(x[i]-x[j]);

yy+=f[i];

}

return yy;

}

int main()

{

double x[N],y[N];

double xx,yy;//xx:x点的x值,yy:x点的y值

int i,n;

printf("请输入函数表中xi的值：");

for(i=0;i<N;i++)

scanf("%lf",&x[i]);

printf("\n请输入函数表中yi的值：");

for(i=0;i<N;i++)

scanf("%lf",&y[i]);

while(1)

{

printf("\n请输入多项式中要求函数近似值的点x的值:xx=");

scanf("%lf",&xx);

printf("\n请输入需要进行拉格朗日插值的次数n：");

scanf("%d",&n);

if(n>=N||n<=0)

return 1;

yy=Lagrange(x,y,xx,n);

printf("\n经过%d次拉格朗日插值后x=%lf处的函数近似值为y=%lf\n",n,xx,yy);

}

return 0;

}

**牛顿插值代码**

#include "stdio.h"

#define N 5//xi的个数

void Difference(double x[],double y[],int n)

//xx:x点的x值，n：表示n次牛顿插值

{

int i,j;

double f[N];

for(i=1;i<=n;i++)

{

f[0]=y[i];

for(j=0;j<i;j++)

if(j!=i)

f[j+1]=(f[j]-y[j])/(x[i]-x[j]);

y[i]=f[i];

}

}

int main()

{

double x[N],y[N];

double xx,yy;//xx:x点的x值,yy:x点的y值

int i,n;

printf("请输入函数表中xi的值：");

for(i=0;i<N;i++)

scanf("%lf",&x[i]);

printf("\n请输入函数表中yi的值：");

for(i=0;i<N;i++)

scanf("%lf",&y[i]);

printf("\n请输入多项式中要求函数近似值的点x的值:xx=");

scanf("%lf",&xx);

printf("\n请输入需要进行牛顿插值的次数n：");

scanf("%d",&n);

if(n>=N||n<=0)

return 1;

Difference(x,y,n);

for(i=n-1;i>=0;i--)

yy=yy\*(xx-x[i])+y[i];

printf("\n经过%d次牛顿插值后x=%lf处的函数近似值为y=%lf\n",n,xx,yy);

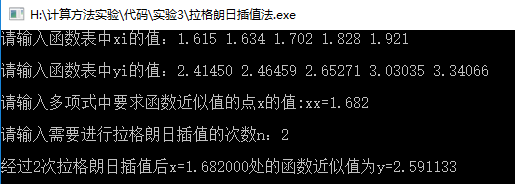
return 0;

}

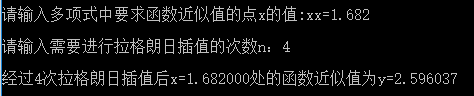
**六、实验结果与分析**

(1)拉格朗日插值

**求f(1.682)**

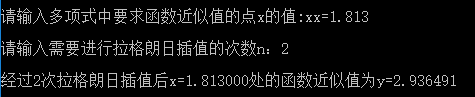


经过二次拉格朗日插值，f(1.682)=2.591133

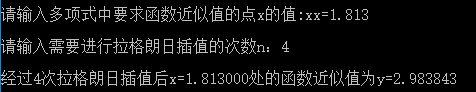


经过四次拉格朗日插值，f(1.682)=2.596037

**求f(1.813)**



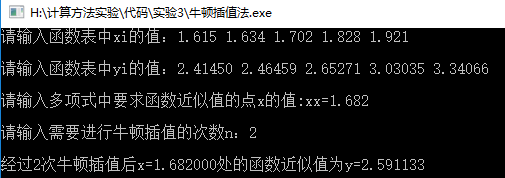
经过二次拉格朗日插值，f(1.813)=2.936491



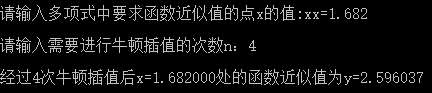
经过四次拉格朗日插值，f(1.813)=2.983843

1. 牛顿插值

**求f(1.682)**

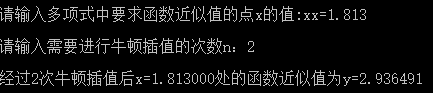


经过二次牛顿插值，f(1.682)=2.591133

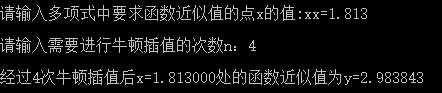


经过四次牛顿插值，f(1.682)=2.596037

**求f(1.813)**



经过二次牛顿插值，f(1.813)=2.936491



经过四次牛顿插值，f(1.813)=2.983843

**分析：**牛顿插值和拉格朗日插值，得到的结果一样，精度相同。

**七、实验心得**

这次实验，让我对牛顿插值和拉格朗日插值，有了更进一步的了解，根据书上的案例以及方法流程，逐渐清楚了如何去处理这类问题。

实验项目名称： 曲线拟合 实验学时： 2

同组学生姓名： 实验地点： 1514

实验日期： 12.19 实验成绩：

批改教师： 批改时间：

一、实验目的

1．掌握最小二乘原理，正规方程组，超定方程组概念。

2．掌握用最小二乘法拟合曲线，超定方程级的最小二乘解。

3．掌握用最小二乘法拟合曲线。

二、实验要求

1．上机前作好充分准备，复习最小二乘拟合方法。

2．上机时要遵守实验室的规章制度，爱护实验设备。

3．记录调试过程及结果，记录并比较与手工运算结果的异同。

4．程序调试完后，须由实验辅导教师在机器上检查运行结果。

5．给出本章实验单元的实验报告。

三、实验设备、环境

1．硬件设备：IBM PC以上计算机，有硬盘和一个软驱、单机和网络环境均可。

2．软件环境： C语言运行环境。

四、实验内容

1．设有如下实验数据

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 1.36 | 1.49 | 1.73 | 1.81 | 1.95 | 2.16 | 2.28 | 2.48 |
| y | 14.094 | 15.069 | 16.844 | 17.378 | 18.435 | 19.949 | 20.963 | 22.495 |

试用最小二乘法分别求一次及二次多项式曲线拟合以上数据。

2．编写程序，分析运行结果。

1. **实验代码**

#include <stdio.h>

#include <math.h>

#define Max 100 //最大数据点的个数

#define M 3 //正规方程组的阶数

void ColPivot(double A[M][M],double B[],int n)

{//列主元高斯消去法求解线性方程组

int i,j,k,m\_i;

double m\_x,temp;

for(i=0;i<n-1;i++)

{ //列主元

j=i+1; m\_i=i; m\_x=fabs(A[i][i]);

for(;j<n;j++)

if(fabs(A[j][i]>m\_x)) //找主元素

{

m\_i=j;

m\_x=fabs(A[j][i]);

}

if(i<m\_i) //交换两行

{

temp=B[i]; B[i]=B[m\_i]; B[m\_i]=temp;

for(j=i;j<n;j++)

{

temp=A[i][j];

A[i][j]=A[m\_i][j];

A[m\_i][j]=temp;

}

}

//消元

for(j=i+1;j<M;j++)

{

temp=-A[j][i]/A[i][i];

B[j]+=B[i]\*temp;

for(k=i;k<M;k++)

A[j][k]+=A[i][k]\*temp;

}

}

}

int main()

{

int i,j,k,n;

double x[Max],y[Max],b[M],a[M][M],c[M];

printf("\n请输入实验数据数n:"); //输入实验数据数n

do

{

scanf("%d",&n);

if(n>Max)

printf("\n输入数据不符合要求，重新输入（0<n<Max）：");

}while(n>Max||n<=0);

printf("Input x[i],i=0,...%d:\n",n-1);

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

scanf("%lf",&x[i]);

printf("Input y[i],i=0,...%d:\n",n-1);

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

scanf("%lf",&y[i]);

for(i=0;i<M;i++) //构造正规方程组

{

for(j=0;j<M;j++)

{

a[i][j]=0;

b[i]=0;

for(k=0;k<n;k++)

{

a[i][j]=a[i][j]+pow(x[k],i+j);

b[i]=b[i]+pow(x[k],i)\*y[k];

}

}

}

ColPivot(a,b,M); //调用列主元消去法函数计算方程组的解

c[M-1]=b[M-1]/a[M-1][M-1];

for(i=M-2;i>=0;i--)

{

c[i]=b[i];

for(j=i+1;j<M;j++)

c[i]-=a[i][j]\*c[j];

c[i]/=a[i][i];

}

printf("Solve is :\n");

for(i=0;i<M;i++)

printf("c[%d]=%lf\n",i,c[i]);

printf("最小二乘法拟合的曲线为：y=%lf+%lfx+%lfx^2",c[0],c[1],c[2]);

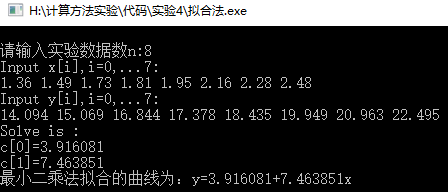
getch();

return 0;

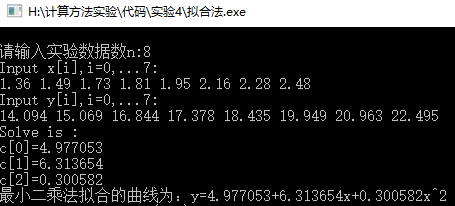
}

1. **实验结果与分析**

最小二乘法一次多项式曲线拟合：



最小二乘法二次多项式曲线拟合



**七、实验心得**

经过这次实验，逐渐了解了最小二乘原理，知道了正规方程组的概念以及超定方程组概念。知道如何使用最小二乘法拟合曲线来解决实际问题，看到拟合结果在我们的编程下被很容易的求解得到，很开心，但编程的过程遇到的问题，一步步的算法，却是最重要的，很开心有这么个体验，相信自己在以后的学习中可以更加独立，实现自己想要的！