

北京化工大学信息科学与技术学院

计算方法
上机实验指导书

程 勇

北京化工大学信息科学与技术学院

二零一六年二月

计算方法上机实验指导书

程 勇

北京化工大学信息科学与技术学院

二零一六年二月

目 录

实验一 LAGRANGE 插值方法.....	4
一 实验目的	4
二 实验原理	4
三 实验环境	4
四 实验内容	4
五 样例程序	5
六 运行结果	5
七 实验报告	6
实验二 NEWTON-COTES 方法.....	7
一 实验目的	7
二 实验原理	7
三 实验环境	7
四 实验内容	7
五 样例程序	8
六 运行结果	8
七 实验报告	8
实验三 解线性方程组的迭代法.....	9
一 实验目的	9
二 实验原理	9
三 实验环境	9
四 实验内容	9
五 样例程序	10
六 实验结果	11
七 实验报告	11
实验四 线性方程组的矩阵分解法.....	12
一 实验目的	12
二 实验原理	12
三 实验环境	12
四 实验内容	12
五 样例程序	13
六 运行结果	14
七 实验报告	14

实验一 Lagrange 插值方法

一 实验目的

1. 熟悉简单的一阶和二阶 Lagrange 插值方法;
2. 学会计算 Lagrange 基函数;
3. 正确构造插值多项式;
4. 对插值结果进行合理分析;

二 实验原理

Lagrange 插值多项式: $L_n(x) = \sum_{i=0}^n l_i(x)y_i$, 其中 $l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)}$

三 实验环境

Windows XP + C

四 实验内容

1. 已知 $\cos \frac{\pi}{6} = \frac{\sqrt{3}}{2}$, $\cos \frac{\pi}{4} = \frac{\sqrt{2}}{2}$, $\cos \frac{\pi}{3} = \frac{1}{2}$, $\cos \frac{\pi}{2} = 0$; 使用 Lagrange 插值多项


式计算 $\cos 47^\circ, \cos 53^\circ, \cos 79^\circ \cos 174^\circ$, 并给出插值多项式;

2. 修改程序直至运行成功, 查看运行结果;

五 样例程序

```
#include <stdio.h>
#include <math.h>
#define MAXSIZE 50
void input(double x[MAXSIZE], double y[MAXSIZE], long n);
void main(void) {
    double x[MAXSIZE], y[MAXSIZE], _x, _y, t;
    long n, i, j;
    printf("\n请输入插值节点的个数: ");
    scanf("%ld", &n);
    input(x, y, n);
    printf("\n请输入插值点: ");
    scanf("%lf", &_x);
    _y = 0;
    for(i = 0; i <= n - 1; i++){
        t = 1;
        for(j = 0; j <= n - 1; j++){
            if(j != i)
                t*=(x-x[j])/(x[i] - x[j]);
            _y+= t*y[i];
        }
    }
    printf("\n插值点(x,y) = (%lf, %lf)。", _x, _y);
}
void input(double x[MAXSIZE], double y[MAXSIZE], long n){
    long i;
    for(i = 0; i <= n - 1; i++){
        printf("\n请输入插值节点x[%ld], y[%ld]:", i, i);
        scanf("%lf, %lf", &x[i], &y[i]);
    }
}
```

六 运行结果



```
C:\Windows\system32\cmd.exe
请输入插值节点的个数: 3
请输入插值节点x[0], y[0]:2.56, 1.6
请输入插值节点x[1], y[1]:2.89, 1.7
请输入插值节点x[2], y[2]:3.24, 1.8
请输入插值点: 3
插值点(x,y) = (3.000000, 1.732101)。请按任意键继续. . .
```

七 实验报告

实验目的明确，步骤清晰，有主要算法，实验数据，实验结果。

实验二 Newton-Cotes 方法

一 实验目的

1. 掌握 Newton-Cotes 算法;
2. 要求程序不断加密对积分区间的等分, 自动地控制 Newton-Cotes 算法中的加速收敛过程;
3. 编写程序, 分析实验结果;

二 实验原理

常见教材 P67-69 页

三 实验环境

Windows XP + C

四 实验内容

计算积分

$$\int_0^{10} \sin x dx$$

五 样例程序

```
#include <stdio.h>
#define MAXSIZE 7
void input(double f[MAXSIZE+1], double a, double b, long n);
void main(void) {
    long c[MAXSIZE][MAXSIZE/2+2] = {{2, 1}, {6, 1, 4}, {8, 1, 3}, {90, 7, 32, 12},
    {288, 19, 75, 50}, {840, 41, 216, 27, 272}, {17280, 751, 3577, 1323, 2989}};
    double a, b, f[MAXSIZE+1], integral;
    long n, i;
    printf("\n请输入积分区间边界a, b:"); scanf("%lf, %lf", &a, &b);
    printf("\n请输入积分节点的个数(2~8):"); scanf("%ld", &n);
    input(f, a, b, n);
    integral = 0;
    for(i = 0; i < n/2; i++)
        integral += (f[i]+f[n-i-1])*c[n-2][i+1]/c[n-2][0];
    if(n%2)
        integral += f[n/2]*c[n-2][n/2+1]/c[n-2][0];
    integral *= b-a;
    printf("\n积分值为=%lf", integral);
}
void input(double f[MAXSIZE+1], double a, double b, long n){
    long i;
    double h;
    h = (b - a) / (n - 1);
    printf("\n请输入求积节点纵坐标:");
    for(i = 0; i <= n-1; i++){
        printf("\nx[%ld] = %lf, f[%ld] = ", i, a+i*h, i);
        scanf("%lf", &f[i]);
    }
}
```

六 运行结果



七 实验报告

实验目的明确，步骤清晰，有主要算法，实验数据，实验结果。

实验三 解线性方程组的迭代法

一 实验目的

1. 掌握雅可比迭代和 Seidel 迭代来求解方程组；
2. 掌握常用的几种迭代格式；
3. 编写程序实现上述迭代方法；
4. 分析实验结果，并估计误差；

二 实验原理

有如下线性方程组 $Ax = b$ 如下：

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

使用迭代法进行求解，主要迭代方法为雅可比迭代和 Gauss-Seidel 迭代，分别见教材 P141 和 P145 页。

三 实验环境

Windows XP + C

四 实验内容

使用 Jacob 迭代法求解下列方程组：

$$\begin{cases} 10x_1 - x_2 - 2x_3 = 7.2 \\ -x_1 + 10x_2 - 2x_3 = 8.3 \\ -x_1 - x_2 + 5x_3 = 4.2 \end{cases}$$

五 样例程序

```
#include <stdio.h>
#include <math.h>
#define MAXSIZE 50
void input(double a[MAXSIZE][MAXSIZE],double b[MAXSIZE],long n);
void output(double x[MAXSIZE],long n);
void main(void){
    double a[MAXSIZE][MAXSIZE],b[MAXSIZE],x[MAXSIZE];
    double epsilon,e,s,oldx;
    long n,i,j,k,maxk;
    printf("\n请输入原方程组的阶数: "); scanf("%ld",&n);
    input(a,b,n);
    printf("\n请输入迭代初始向量: ");
    for(i=0;i<=n-1;i++) scanf("%lf",&x[i]);
    printf("\n请输入最大迭代次数: ");
    scanf("%ld",&maxk);
    printf("\n请输入误差上限: ");
    scanf("%lf",&epsilon);
    for(k=1;k<=maxk;k++) {
        e=0;
        for(i=0;i<=n-1;i++) {
            oldx=x[i]; s=0;
            for(j=0;j<=n-1;j++)
                if(j!=i) s+=a[i][j]*x[j];
            x[i]=(b[i]-s)/a[i][i];
            if(e<fabs(oldx-x[i]))
                e=fabs(oldx-x[i]);
        }
        if(e<epsilon) break;
    }
    if(k<=maxk)
        output(x,n);
    else
        printf("\n迭代次数已超过上限。");
}
```

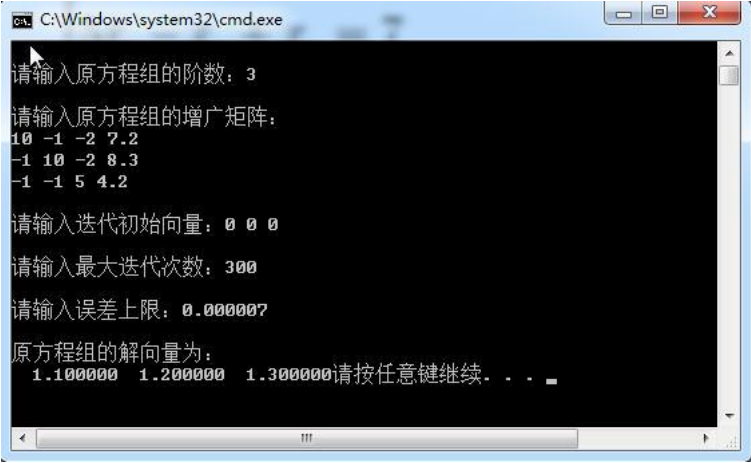
```

void input(double a[MAXSIZE][MAXSIZE],double b[MAXSIZE],long n) {
    long i,j;
    printf("\n请输入原方程组的增广矩阵: \n");
    for(i=0;i<=n-1;i++) {
        for(j=0;j<=n-1;j++) scanf("%lf",&a[i][j]);
        scanf("%lf",&b[i]);
    }
}

void output(double x[MAXSIZE],long n){
    long i;
    printf("\n原方程组的解向量为: \n");
    for(i=0;i<=n-1;i++)
        printf(" %lf",x[i]);
}

```

六 实验结果



```

C:\Windows\system32\cmd.exe
请输入原方程组的阶数: 3
请输入原方程组的增广矩阵:
10 -1 -2 7.2
-1 10 -2 8.3
-1 -1 5 4.2
请输入迭代初始向量: 0 0 0
请输入最大迭代次数: 300
请输入误差上限: 0.000007
原方程组的解向量为:
1.100000 1.200000 1.300000请按任意键继续. . .

```

七 实验报告

实验目的明确，步骤清晰，有主要算法，实验数据，实验结果。

实验四 线性方程组的矩阵分解法

一 实验目的

1. 掌握采用矩阵 LU 分解方法来求解线性方程组；
2. 编程实现矩阵 LU 分解算法；

二 实验原理

具体参见教材 P177-180 页。

三 实验环境

Windows XP + C

四 实验内容

1. 写出矩阵 LU 分解法解线性方程组算法，编一程序上机调试出结果，要求所编程序适用于任何一解线性方程组问题，即能解决这一类问题，而不是某一个具体问题。
2. 使用矩阵 LU 分解法求解下列方程组：

$$\begin{cases} 10x_1 - x_2 - 2x_3 = 7.2 \\ -x_1 + 10x_2 - 2x_3 = 8.3 \\ -x_1 - x_2 + 5x_3 = 4.2 \end{cases}$$

五 样例程序

```
#include <stdio.h>
#include <math.h>
#define MAXSIZE 50
void input(double a[MAXSIZE][MAXSIZE],double x[MAXSIZE],long n);
void output(double x[MAXSIZE],long n);
void main(void) {
    double a[MAXSIZE][MAXSIZE],x[MAXSIZE],s;
    long n,i,j,k;
    printf("\n请输入原方程组的阶数: "); scanf("%ld",&n);
    input(a,x,n);
    for(k=0;k<=n-2;k++) {
        for(i=k+1;i<=n-1;i++) {
            s=0;
            for(j=0;j<=k-1;j++) s+=a[i][j]*a[j][k];
            a[i][k]=(a[i][k]-s)/a[k][k];
        }
        for(j=k+1;j<=n-1;j++){
            s=0;
            for(i=0;i<=k;i++) s+=a[k+1][i]*a[i][j];
            a[k+1][j]=-s;
        }
    }
    for(i=1;i<=n-1;i++){
        s=0;
        for(j=0;j<=i-1;j++) s+=a[i][j]*x[j];
        x[i]=-s; }
    for(i=n-1;i>=0;i--){
        s=0;
        for(j=i+1;j<=n-1;j++) s+=a[i][j]*x[j];
        x[i]=(x[i]-s)/a[i][i];
    }
    output(x,n);
}
```


```

void input(double a[MAXSIZE][MAXSIZE],double x[MAXSIZE],long n) {
    long i,j;
    printf("\n请输入原方程组的增广矩阵: \n");
    for(i=0;i<=n-1;i++) {
        for(j=0;j<=n-1;j++) scanf("%lf",&a[i][j]);
        scanf("%lf",&x[i]);
    }
}

void output(double x[MAXSIZE],long n) {
    long i;
    printf("\n原方程组的解为: \n");
    for(i=0;i<=n-1;i++) printf(" %lf",x[i]);
}

```

六 运行结果



```

C:\Windows\system32\cmd.exe
请输入原方程组的阶数: 3
请输入原方程组的增广矩阵:
10 -1 -2 7.2
-1 10 -2 8.3
-1 -1 5 4.2

原方程组的解为:
1.100000 1.200000 1.300000请按任意键继续. . .

```

七 实验报告

实验目的明确，步骤清晰，有主要算法，实验数据，实验结果。