

## 8.4 用列主元消去法解线性方程组

### 1. 例题与程序

解方程组

$$\begin{pmatrix} 0.0001 & 0.5402 & 0.3425 \\ 1.235 & 2.567 & 0.9750 \\ 1.024 & 2.001 & 4.555 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.8828 \\ 4.777 \\ 7.580 \end{pmatrix}$$

Visual Studio 2012 程序编辑如下：

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

void scanarray(double **v, int n);
int Gauss(double **v, double *solution, int n);

void main() /*主函数，其作用在于输入和输出，以及为数组分配内存空间*/
{
    clock_t start, finish;
    int i, n, count;
    double *m, **p, dauration;
    printf("请输入线性方程组的未知数的个数:");
    scanf("%d", &n); /*接收用户输入的线性方程组的阶数，以此来决定如何分配内存空间*/

    p=(double **)malloc(n*sizeof(double *)); /*malloc 函数用于创建动态二维数组*/
    for(i=0; i<n; i++)
        p[i]=(double *)malloc((n+1)*sizeof(double)); /*生成一个 n 行 n+1 列的增广矩阵数组*/
    scanarray(p, n); /*调用 scanarray 函数来为二维数组录入用户的数据*/

    m=(double *)malloc(n*sizeof(double)); /*调用 malloc 函数创建动态数组以保存结果*/

    start=clock();
    count=Gauss(p, m, n); /*调用 Gauss 函数计算方程的解的值，返回值为 1 说明系数矩阵奇异*/
    finish=clock();
```

```

        if(count!=0)
            printf("\n 系数矩阵奇异\n");
        else
        {
            printf("\n 线性方程组的解为:\n");
            for(i=0;i<n;i++)
            {
                printf("x(%d)=%lf\n", i, *(m+i));
            }

        }
        dauration=(double) (finish-start);
        printf("\n 运行时间 (毫秒) :%lf", dauration);
        system("pause");
    }

void scanarray(double **v, int n)                                /*scanarray 函数作用是
从用户那里接收行列式的值*/
{
    printf("请输入线性方程组增广矩阵的元素 (输入一行后请按回车, 每个元素间空一
格) :\n");

    int i, j;
    for(i=0; i<n; i++)
        for(j=0; j<n+1; j++)
            scanf("%lf", &v[i][j]);
}

int Gauss(double **v, double *solution, int n)                 /*Gauss 函数为线性方
程组计算函数*/
{
    int i, j, k, flag=0, p=0;
    double m, s, temp, d;
    for(k=0; k<n&&flag!=1; k++)                                /* k 的作用在于记录当
前在从上往下数第几个主对角线元素*/
    {
        p=k;
        d=0;
        for(j=k; j<n; j++)
        {
            d=d+ (*(v+k)+j));
        }
        if(d==0)

```

```

        {
            flag=1;                                /*发现矩阵奇异（化简中
出现一行全为零），以后不再进行行变换。*/
        }

        temp=fabs((*(*(v+k)+k)));
        for(i=k+1;i<n&&flag!=1;i++)                /*利用循环选出列主元
素*/
        {
            if(fabs((*(*(v+i)+k)))>temp)
            {
                p=i;
                temp=fabs((*(*(v+i)+k)));
            }
        }

        if(temp==0)                                /*发现矩阵奇异（对角
线有零元素）*/
            flag=1;
        else if(p<n&&p!=k)
            for(j=k;j<n+1;j++)                    /*对矩阵进行初等行变
化*/
            {
                s=(*(*(v+k)+j));
                (*(*(v+k)+j))=(*(*(v+p)+j));
                (*(*(v+p)+j))=s;
            }

        for(i=k+1;i<n&&flag!=1;i++)                /*对第 i 行进行操作来
消元, 其中 i 的初始值为 k+1, 终止值为行列式阶数*/
        {
            m=(*(*(v+i)+k))/(*(*(v+k)+k));        /*m 为高斯消元法中的
乘子*/
            for(j=k;j<n+1&&flag!=1;j++)
            {
                (*(*(v+i)+j))=(*(*(v+i)+j))-m*(*(*(v+k)+j)); /*对 i 行中的 j 列元素进
行运算, j 初始值为 k, 终止值为行列式阶数*/
            }
        }
    }
}

```

```

printf("正向消元，矩阵被化为:\n");
for(i=0;i<n;i++)
{
    for(j=0;j<n+1;j++)
        printf("%lf ",v[i][j]);
    printf("\n");
}

```

/\*回带过程（反向消元）\*/

```

if(flag==0)
{

    for(k=n-1;k>=0;k--)
    {

        for(i=k-1;i>=0;i--)
        {
            m=*(v+i+k)/(v[k]);
            for(j=n;j>=0;j--)
            {
                *(v+i+j)=*(v+i+j)-m*(v[k+j]);
            }
        }
    }
}

```

```

printf("反向消元，矩阵被化为:\n");
for(i=0;i<n;i++)
{
    for(j=0;j<n+1;j++)
        printf("%lf ",v[i][j]);
    printf("\n");
}

```

```

        for(k=n-1;k>=0;k--)                                /*单位化*/
        {
            *((v+k)+n)=*((v+k)+n)/(*((v+k)+k));
            *((v+k)+k)=1;
        }

printf("单位化后，矩阵被化为:\n");
for(i=0;i<n;i++)
{
    for(j=0;j<n+1;j++)
        printf("%lf ",v[i][j]);
    printf("\n");
}

}

/*结果输出部分*/

for(k=0;k<n;k++)
{
    *(solution+k)=*((v+k)+n);
}

return flag;

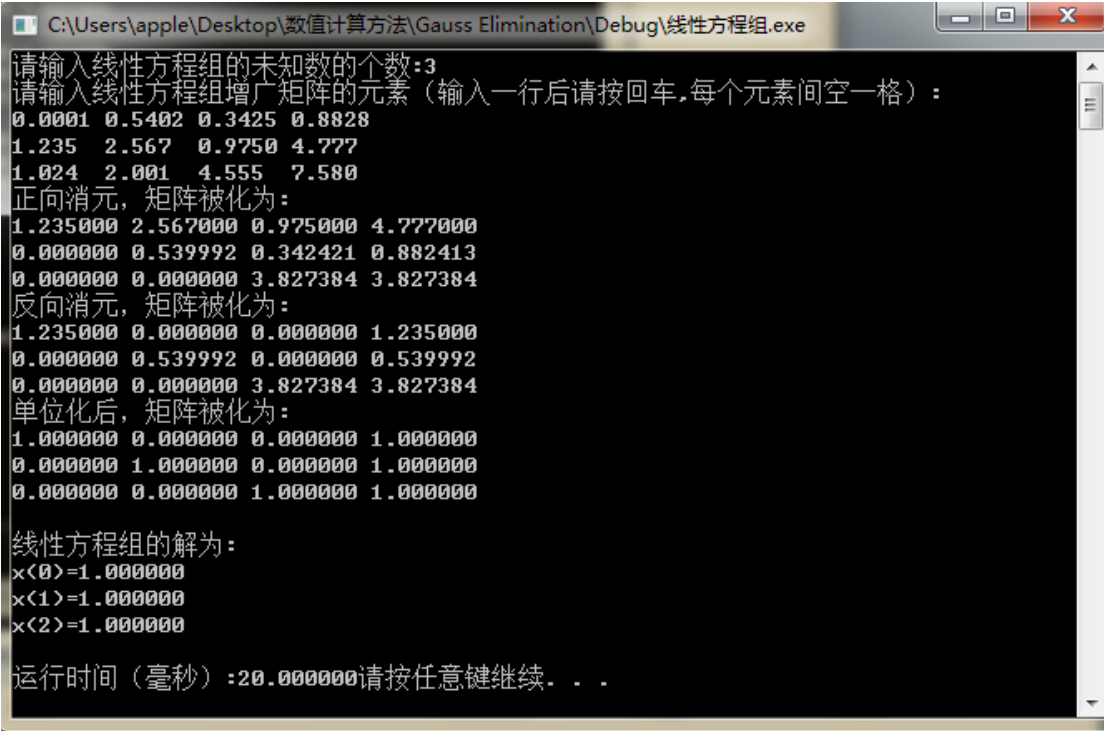
}

```

VC++程序运行结果下:

在生成的 exe 文件中输入线性方程组的增广矩阵

$$\begin{pmatrix} 0.0001 & 0.5402 & 0.3425 & 0.8828 \\ 1.235 & 2.567 & 0.9750 & 4.777 \\ 1.024 & 2.001 & 4.555 & 7.580 \end{pmatrix}$$



```
C:\Users\apple\Desktop\数值计算方法\Gauss Elimination\Debug\线性方程组.exe
请输入线性方程组的未知数的个数:3
请输入线性方程组增广矩阵的元素(输入一行后请按回车,每个元素间空一格):
0.0001 0.5402 0.3425 0.8828
1.235 2.567 0.9750 4.777
1.024 2.001 4.555 7.580
正向消元,矩阵被化为:
1.235000 2.567000 0.975000 4.777000
0.000000 0.539992 0.342421 0.882413
0.000000 0.000000 3.827384 3.827384
反向消元,矩阵被化为:
1.235000 0.000000 0.000000 1.235000
0.000000 0.539992 0.000000 0.539992
0.000000 0.000000 3.827384 3.827384
单位化后,矩阵被化为:
1.000000 0.000000 0.000000 1.000000
0.000000 1.000000 0.000000 1.000000
0.000000 0.000000 1.000000 1.000000
线性方程组的解为:
x<0>=1.000000
x<1>=1.000000
x<2>=1.000000
运行时间(毫秒):20.000000请按任意键继续. . .
```

用数学软件 MATLAB 对结果进行检验:

```
>> A=[0.0001 0.5402 0.3425;1.235 2.567 0.9750;1.024 2.001 4.555]
```

A =

```
0.0001    0.5402    0.3425
1.2350    2.5670    0.9750
1.0240    2.0010    4.5550
```

```
>> b=[0.8828 ;4.777 ;7.580]
```

b =

```
0.8828
4.7770
7.5800
```

7.5800

```
>> X=inv(A)*b
```

X =

1.0000

1.0000

1.0000

对比结果，发现数学软件解出的结果于所编程序解出的结果相同，程序可靠。Gauss 消元运行时间为 20ms，较短。并且从程序输出的消元中间过程看出，该程序确实进行了选列主元的操作。

## 2. 实验题

(1) 解方程组

$$\begin{pmatrix} 0.101 & 2.304 & 3.555 \\ -1.347 & 3.712 & 4.632 \\ -2.835 & 1.072 & 5.643 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1.183 \\ 2.137 \\ 3.035 \end{pmatrix}$$

VC++运行结果:

在运行生成的 exe 文件输入原方程组的增广矩阵:

$$\begin{pmatrix} 0.101 & 2.304 & 3.555 & 1.183 \\ -1.347 & 3.712 & 4.623 & 2.137 \\ -2.835 & 1.072 & 5.643 & 3.035 \end{pmatrix}$$

```
C:\Users\apple\Desktop\数值计算方法\Gauss Elimination\Debug\线性方程组.exe
请输入线性方程组的未知数的个数:3
请输入线性方程组增广矩阵的元素(输入一行后请按回车,每个元素间空一格):
0.101 2.304 3.555 1.183
-1.347 3.712 4.623 2.137
-2.835 1.072 5.643 3.035
正向消元, 矩阵被化为:
-2.835000 1.072000 5.643000 3.035000
0.000000 3.202658 1.941829 0.694974
0.000000 0.000000 2.335926 0.782872
反向消元, 矩阵被化为:
-2.835000 0.000000 0.000000 1.128993
0.000000 3.202658 0.000000 0.044181
0.000000 0.000000 2.335926 0.782872
单位化后, 矩阵被化为:
1.000000 0.000000 0.000000 -0.398234
0.000000 1.000000 0.000000 0.013795
0.000000 0.000000 1.000000 0.335144

线性方程组的解为:
x<0>=-0.398234
x<1>=0.013795
x<2>=0.335144

运行时间(毫秒):30.000000请按任意键继续. . .
```

使用数学软件 MATLAB 对结果进行检验:

```
>> A=[0.101 2.304 3.555;-1.347 3.712 4.623;-2.835 1.072 5.643]
```

A =

```
0.101000000000000    2.304000000000000    3.555000000000000
-1.347000000000000    3.712000000000000    4.623000000000000
-2.835000000000000    1.072000000000000    5.643000000000000
```

```
>> b=[1.183;2.137;3.035]
```

b =

```
1.183000000000000
2.137000000000000
3.035000000000000
```

```
>> X=inv(A)*b
```

X =

```
-0.39823376874172
0.01379506599727
0.33514424151482
```



>>

对比所编程序与数学软件 MATLAB 所给的结果,发现所编程序计算出的结果可靠。程序主要模块 Gauss 消元模块运行时间为 30ms,较短。但比例题运行时间多 10ms。从程序的中间结果中看出程序的消元过程进行了选列主元操作。

### 程序健壮性测试:

输入一个奇异的线性方程组:(典型奇异矩阵:一二行的系数相同)



```
C:\Users\apple\Desktop\数值计算方法\Gauss Elimination\Debug\线性方程组.exe
请输入线性方程组的未知数的个数:4
请输入线性方程组增广矩阵的元素(输入一行后请按回车,每个元素间空一格):
1 2 3 4 5
1 2 3 4 20
2 5 2 4 29
3 4 2 1 20
正向消元,矩阵被化为:
3.000000 4.000000 2.000000 1.000000 20.000000
0.000000 2.333333 0.666667 3.333333 15.666667
0.000000 -0.000000 2.142857 2.714286 8.857143
0.000000 -0.000000 0.000000 0.000000 -15.000000
系数矩阵奇异
运行时间(毫秒):20.000000请按任意键继续. . .
```

程序识别出了奇异矩阵,并且中断了后面的反向消元(回带)运算。达到了预想的目的。

### 实验总结:

选列主元高斯消元法,尽可能的避免了消元过程中小数做除数的情况,使得结果的误差减小,特别是在原线性方程组系数矩阵对角线元素很小时候,该算法比较有效。所编程序全部使用指针进行操作,并且程序坚持模块化的思想编辑,模块包括主函数模块,其功能在于分配内存空间,调用计算函数,输出结果等。而子函数有 Gauss 函数,列主元 gauss 消元法实现模块。录入数据时采用子函数录入,便于用户使用生成的 exe 时数据录入。并且在系数矩阵奇异时候程序能够识别出来,健壮性较好。