

## 3J 数値解析 実習課題 No.4 2017/7/31

3J 31 堀井萌希

### 0.実施環境

課題の作成・実行に用いたソフトウェア環境を表1に示す。

表1:実施に用いたソフトウェア環境

OS	Windows7
C コンパイラ	CodeBlocks

### 1. ガウスの消去法のプログラムの作成

以下に各プログラムのソースと実行結果を示す。

#### 1.1 ソース

```
#include <stdio.h>
#define N 4

int main(void)
{

    double a[N][N+1]={2.0,-4.0,3.0,-1.0,-2.0},
    {1.0,-2.0,2.0,1.0,1.0},
    {1.0,-5.0,4.0,-3.0,-8.0},
    {3.0,2.0,-2.0,-2.0,1.0}};
    int i,j,k,l,pivot;
    double x[N];
    double p,q,m,b[1][N+1];

    for(i=0;i<N;i++) {
        m=0;
        pivot=i;

        for(l=i;l<N;l++) {

            if((a[l][i])>m) {
                m=a[l][i];
```

```

                                pivot=l;
                                }
                                }

    if(pivot!=i) {
        for(j=0;j<N+1;j++) {
            b[0][j]=a[i][j];
            a[i][j]=a[pivot][j];
            a[pivot][j]=b[0][j];
        }
    }
}

for(k=0;k<N;k++) {
    p=a[k][k];
    a[k][k]=1;

    for(j=k+1;j<N+1;j++) {
        a[k][j]/=p;
    }

    for(i=k+1;i<N;i++) {
        q=a[i][k];

        for(j=k+1;j<N+1;j++) {
            a[i][j]-=q*a[k][j];
        }

        a[i][k]=0;
    }
}

for(i=N-1;i>=0;i--) {
    x[i]=a[i][N];
    for(j=N-1;j>i;j--) {
        x[i]-=a[i][j]*x[j];
    }
}

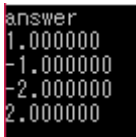
```

```

        }
    }
    printf("answer¥n");
    for(i=0;i<N;i++) {
        printf("%f¥n",x[i]);
    }
    return 0;
}

```

## 1.2 結果



```

answer
1.000000
-1.000000
-2.000000
2.000000

```

## 1.3 考察

手計算で行ったときと同様の結果が得られた。

## 2. LU 分解法のプログラムの作成

以下に各プログラムのソースと実行結果を示す。

### 2.1 ソース

```

#include<stdio.h>

#define n 3

int main(void){
    int i,j,k;

    double a[n][n]={4,1,1,
                    {1,4,1},
                    {1,1,4}};

    double b[n]={1,2,3};
    double u[n][n],l[n][n],x[n],y[n];
    double bly = 0;
    double yux = 0;

    for(i=0;i<n;i++){
        for( j = 0; j < n; j++ ){
            if( i == j ){ l[i][j] = 1;

```

```

        else{ l[i][j] = 0;
        }
        u[i][j] = a[i][j];
    }

}

for( i = 0; i < n; i++ ){
    for( j = i+1; j < n; j++ ){
        l[j][i] = u[j][i] / u[i][i];

        for( k = i+1; k < n; k++ ){
            u[j][k] = u[j][k] - u[i][k] * l[j][i];

        }
    }
}

printf("下三角行列  L¥n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("l[%d][%d]%%lf",j,i,l[j][i]);
    }
    printf("¥n");
}

printf("上三角行列  U¥n");
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        printf("u[%d][%d]%%lf",j,i,u[j][i]);
    }
    printf("¥n");
}

for(i=0;i<n;i++){
    y[i]=b[i];
}

for(i=0;i<n;i++){
    y[i]/=l[i][i];
    for(j=i+1;j<n;j++){

```

```

        y[j]-=y[i]*l[j][i];
    }
}
for(i=0;i<n;i++){
    x[i]=y[i];
}
for(i=n-1;i>=0;i--){
    x[i]/=u[i][i];
    for(j=i-1;j>=0;j--){
        x[j]-=x[i]*u[j][i];
    }
}
printf("解¥n");
printf("x=%lf",x[0]);
printf("y=%lf",x[1]);
printf("z=%lf",x[2]);

return 0;
}

```

## 2.2 結果

```

下三角行列 L
l[0][0]1.000000l[1][0]0.250000l[2][0]0.250000
l[0][1]0.000000l[1][1]1.000000l[2][1]0.200000
l[0][2]0.000000l[1][2]0.000000l[2][2]1.000000
上三角行列 U
u[0][0]4.000000u[1][0]1.000000u[2][0]1.000000
u[0][1]1.000000u[1][1]3.750000u[2][1]0.750000
u[0][2]1.000000u[1][2]0.750000u[2][2]3.600000
解
x=0.000000y=0.333333z=0.666667

```

## 2.3 考察

手計算で行ったときと同様の結果が得られた。