

=====

7 月 10 日（金）第 11 回数値解析 I 提出課題 19TM054 浅野 駿介

提出日：2020/07/31

=====

<作成プログラム>

```
#include <stdio.h>
```

```
#define N 4//4 行 4 列の行列の計算
```

```
int count = 0;//計算回数を記録する
```

```
//計算過程を出力する関数
```

```
void print_matrixA(double A[][N + 1]) {
```

```
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N + 1; j++) {
            printf("%lf", A[i][j]);
```

```
        }
        printf(" ¥n");
```

```
    }
    printf(" ¥n");
```

```
}
```

```
//解を出力する関数
```

```
void print_matrix(double X[]) {
```

```
    for (int i = 0; i < N; i++) {
        printf("%lf¥n", X[i]);
```

```
    }
    printf(" ¥n");
```

```
}
```

```
void my_gauss(double A[][N + 1], double X[]) {
```

```
    int i, j, h;
```

```
    //前進消去を行う
```

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

```

        for (j = N; j >= i; j--) {
            A[i][j] = A[i][j] / A[i][i];
            count = count + 1;
        }
        for (h = i + 1; h < N; h++) {
            for (j = N; j >= i; j--) {
                A[h][j] = A[h][j] - A[i][j] * A[h][i];
            }
        }
        print_matrixA(A);
    }
    //後退代入を行う
    X[3] = A[4][5];
    X[2] = A[3][5]-X[3]*A[3][4];
    X[1] = A[2][5]-X[3]*A[2][4]-X[2]*A[2][3];
    X[0] = A[1][5]-X[3]*A[1][4]-X[2]*A[1][3]-X[1]*A[1][2];

    double aih;

    for (i = N - 1; i >= 0; i--) {
        for (h = i - 1; h >= 0; h--) {
            aih = A[h][i];
            for (j = 0; j <= N; j++) {
                A[h][j] = A[h][j] - A[i][j] * aih;
            }
        }
        print_matrixA(A);
        X[i] = A[i][N];
    }
    print_matrix(X);
    printf("Total count=%d¥n", count); //計算回数を表示する
}

void main() {

```

```

double A[N][N + 1] = { {4.0,1.0,-3.0,0.0,0.0},
                        {2.0,0.0,3.0,-2.0,0.0},
                        {0.0,-1.0,3.0,2.0,25.0},
                        {10.0,-4.0,0.0,-3.0,5.0} };//行列式

double X[N] = { 0.0,0.0,0.0,0.0 };//解を入れる行列

print_matrixA(A);
my_gauss(A, X);
}

```

<出力結果>

```

4.000000 1.000000 -3.000000 0.000000 0.000000
2.000000 0.000000 3.000000 -2.000000 0.000000
0.000000 -1.000000 3.000000 2.000000 25.000000
10.000000 -4.000000 0.000000 -3.000000 5.000000

1.000000 0.250000 -0.750000 0.000000 0.000000
0.000000 -0.500000 4.500000 -2.000000 0.000000
0.000000 -1.000000 3.000000 2.000000 25.000000
0.000000 -6.500000 7.500000 -3.000000 5.000000

1.000000 0.250000 -0.750000 0.000000 0.000000
0.000000 1.000000 -9.000000 4.000000 -0.000000
0.000000 0.000000 -6.000000 6.000000 25.000000
0.000000 0.000000 -51.000000 23.000000 5.000000

1.000000 0.250000 -0.750000 0.000000 0.000000
0.000000 1.000000 -9.000000 4.000000 -0.000000
0.000000 0.000000 1.000000 -1.000000 -4.166667
0.000000 0.000000 0.000000 -28.000000 -207.500000

1.000000 0.250000 -0.750000 0.000000 0.000000
0.000000 1.000000 -9.000000 4.000000 -0.000000
0.000000 0.000000 1.000000 -1.000000 -4.166667
0.000000 0.000000 0.000000 1.000000 7.410714

```

1.0000000.250000-0.7500000.0000000.000000
0.0000001.000000-9.0000000.000000-29.642857
0.0000000.0000001.0000000.0000003.244048
0.0000000.0000000.0000001.0000007.410714

1.0000000.2500000.0000000.0000002.433036
0.0000001.0000000.0000000.000000-0.446429
0.0000000.0000001.0000000.0000003.244048
0.0000000.0000000.0000001.0000007.410714

1.0000000.0000000.0000000.0000002.544643
0.0000001.0000000.0000000.000000-0.446429
0.0000000.0000001.0000000.0000003.244048
0.0000000.0000000.0000001.0000007.410714

1.0000000.0000000.0000000.0000002.544643
0.0000001.0000000.0000000.000000-0.446429
0.0000000.0000001.0000000.0000003.244048
0.0000000.0000000.0000001.0000007.410714

2.544643
-0.446429
3.244048
7.410714

Total count=14

<理解した内容、感想、注意点など>

- ・プログラム全体で使う定数の宣言方法が分かった.
- ・手計算だとかなり手間がかかる計算もプログラムを使えばかなり簡単に計算できることが分かった.
- ・前進消去と後退代入で分けたほうがプログラミングを作りやすいことが分かった.
- ・Nの値が大きくなると計算回数は多くなる.