# 수치해석 과제 3

**2017030500 홍승표**

## 실행 방법

- make clean && make && ./runHw03 <데이터 파일 경로>
- 특이값 행렬의 경우, main.c 최상단에서 #define SKIP_GAUSSJ 추가
    - 특이값 행렬에서 gaussj 오류로 인해 프로그램 중지를 막기 위한 macro

```
C main.c M ×
hw03 > C main.c > print_mat(float **, int, int, int, int)
You, 50 seconds ago | 1 author (You)
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "nr.h"
#include "nrutil.h"

// #define SKIP_GAUSSJ
```

## 실행 결과

- 각 파일별로 Gauss-Jordan, LU Decomposition, 특이값분해, Iterative Improvement로 해를 구함
- 각 파일별로 계수행렬의 역행렬과 특이값을 구함

## 파일 1 (lineq1.dat)

```
File: lineq_dat/lineq1.dat
Numerical Recipes run-time error...
gaussj: Singular Matrix
...now exiting to system...
```

- 행렬값이 0인 특이행렬이므로 실행되지 않음.
- Mac M2 Pro 환경에서 NR 폴더의 gaussj.c를 그대로 사용 시, singular matrix가 감지되지 않아서 프로그램이 중단되지 않고 아래와 같은 무의미한 해가 도출됨

```
File: lineq_dat/lineq1.dat
Method: Gauss-Jordan, Solution: [2.000000, -1.000000, -1.000000, -1.000000]
Method: LU Decomposition, Solution: [3.000000, 1.000001, -4.000001, -2.000000]
Method: SVD, Solution: [1.733333, -1.533333, -0.200000, -0.733333]
Method: Improved LU Decomposition, Solution: [3.000000, 1.000001, -4.000001, -2.000000]
Method: Improved LU Decomposition (noisy), Solution: [3.000008, 1.131539, -3.244396, -1.541350]
Method: Improved LU Decomposition (mprove), Solution: [24.541349, 44.082710, -68.624054, -23.541349]
Inverse Matrix:
-33554432.000000        -12582912.000000        12582911.000000         4194304.500000
-67108888.000000        -25165830.000000        25165832.000000         8388610.000000
100663312.000000        37748740.000000         -37748740.000000        -12582915.000000
33554432.000000         12582912.000000         -12582911.000000        -4194304.500000
Determinant: -0.000000
```

- 아래와 같이 특이값 판정 조건문 수정하여 Mac M2 Pro 에서도 특이값 감지가 가능함

```
C main.c    C gaussj.c M ×    lineq1.dat
NRs > ansi > recipes > C gaussj.c > gaussj(float **, int, float **, int)
void gaussj(float **a, int n, float **b, int m) {
                }
            ++(ipiv[icol]);
            if (irow != icol) {
                for (l = 1; l <= n; l++) SWAP(a[irow][l], a[icol][l])
                for (l = 1; l <= m; l++) SWAP(b[irow][l], b[icol][l])
            }
            indxr[i] = irow;
            indxc[i] = icol;
            // 수정 부분
            // if (a[icol][icol] == 0.0) nrerror("gaussj: Singular Matrix");
            if (fabs(a[icol][icol]) < 1e-7) nrerror("gaussj: Singular Matrix");
            pivinv = 1.0 / a[icol][icol];
            a[icol][icol] = 1.0;        You, 3 minutes ago • code done
            for (l = 1; l <= n; l++) a[icol][l] *= pivinv;
            for (l = 1; l <= m; l++) b[icol][l] *= pivinv;
            for (ll = 1; ll <= n; ll++)
```

### 파일 2 (lineq2.dat)

```
File: lineq_dat/lineq3.dat
Method: Gauss-Jordan, Solution: [-0.326608, 1.532292, -1.044825, -1.587448, 2.928480, -2.218931]
Method: LU Decomposition, Solution: [-0.326608, 1.532292, -1.044826, -1.587447, 2.928480, -2.218930]
Method: SVD, Solution: [-0.326608, 1.532290, -1.044823, -1.587447, 2.928478, -2.218929]
Method: Improved LU Decomposition, Solution: [-0.326608, 1.532292, -1.044826, -1.587447, 2.928480, -2.218930]
Method: Improved LU Decomposition (noisy), Solution: [-0.326600, 1.663830, -0.289220, -1.128797, 3.461247, -1.999971]
Method: Improved LU Decomposition (mprove), Solution: [-0.326608, 1.532292, -1.044826, -1.587447, 2.928480, -2.218930]
Inverse Matrix:
-0.162205       0.122801        0.024068        -0.016431       -0.022840       0.046132
0.169407        -0.041117       0.228313        -0.087624       0.180306        -0.395655
-0.011636       0.122745        -0.117407       -0.180981       0.015910        0.186766
0.105669        -0.051726       -0.108916       0.299774        0.000859        -0.190541
-0.053026       -0.042361       0.160508        -0.224034       0.161811        0.015024
-0.062341       -0.064694       -0.234216       0.351126        -0.364828       0.434633
Determinant: 16178.401367
```

### 파일 3 (lineq3.dat)

```
File: lineq_dat/lineq2.dat
Method: Gauss-Jordan, Solution: [-2.873565, -0.612356, 0.976277, 0.635818, -0.553441]
Method: LU Decomposition, Solution: [-2.873566, -0.612357, 0.976277, 0.635819, -0.553441]
Method: SVD, Solution: [-2.873564, -0.612357, 0.976277, 0.635818, -0.553440]
Method: Improved LU Decomposition, Solution: [-2.873566, -0.612357, 0.976277, 0.635819, -0.553441]
Method: Improved LU Decomposition (noisy), Solution: [-2.873558, -0.480819, 1.731883, 1.094469, -0.020674]
Method: Improved LU Decomposition (mprove), Solution: [-2.873566, -0.612357, 0.976277, 0.635818, -0.553441]
Inverse Matrix:
0.354536        0.766945        0.207769        -0.595412       0.253128
0.035454        0.126695        0.195777        -0.159541       0.050313
-0.138686       -0.098540       -0.096715       0.124088        0.016423
-0.052138       -0.303962       -0.023201       0.234619        -0.044578
0.149114        0.459333        0.051356        -0.171011       0.042492
Determinant: 3835.999512
```