수치해석 HW#3

2018008559 컴퓨터소프트웨어학부 신상윤

1. Introduction

1. 목적

가우스 조르단 소거법, LU분해, 특이값 분해를 이용하여 연립 선형방정식을 풀어본다. 더 나아가 iterative improvement를 통해 더 좋은 해를 찾아본다. 마지막으로 행렬의 행렬값, 역행렬을 구해본다.

2. Process

```
int n = 4;
const DP A[n*n]={4,2,3,-1,-2,-1,-2,2,5,3,4,-1,11,4,6,1};
const DP B[n] = {4,-3,4,11};
```

행렬을 읽어서 A,b,n을 입력해준다. 이후에 변경되는 값은 없다.

```
Mat_DP matrix_A1(A,n,n),matrix_B1(B,n,1);

NR::gaussj(matrix_A1,matrix_B1);
cout << "solution by gauss-jordan elimination\n";
for(int i=0; i<n; i++){
    cout << matrix_B1[i][0] << " ";
}
cout << "\n\n";
```

가우스 조르단 소거법을 실행하여 해를 구한다.

```
cout << "inverse matrix of A\n";
for(int i=0; i<n; i++){
    for(int j=0; j<n; j++){
        cout << matrix_A1[i][j] << " ";
    }
    cout << "\n";
}
cout << "\n";
```

가우스 조르단 소거법을 실행한 뒤 matrix A는 A의 inverse가 된다.

```
NR::ludcmp(matrix_A2, indx2,d2);
cout << "after LU decomposition₩n";
for(int i=0; i<n; i++){
    for(int j=0; j<n; j++){
        cout << matrix_A2[i][j] << " ";
    }
    cout << "₩n";
}
cout << "₩n";

cout << "d value :  " << d2 << "₩n₩n";

NR::lubksb(matrix_A2, indx2,matrix_x2);

cout << "solution by LU decomposition₩n";
for(int i=0; i<n; i++){
    cout << matrix_x2[i] << " ";
}
cout << "₩n₩n";
```

LU 분해를 실행한 뒤 LU가 합쳐서 나온 matrix_A를 출력하여 행렬값을 알수있도록한다. LU 분해를 후 구한 해도 출력한다. d2는 행이 얼마나 교환 되었는지를 의미하는데, +1이면 짝수번, -1이면 홀수번 교환되어서 행렬값을 구할 때 참고하면 된다.

```
NR::mprove(matrix_AA2,matrix_A2,indx2,matrix_BB2,matrix_x2);
cout << "after iterative improvement₩n";
for(int i=0; i<n; i++){
    cout << matrix_x2[i] << " ";
}
cout << "₩n₩n";
```

이후 iterative improvement method를 실행하여 더 정밀한 해를 찾아본다.

```
Mat_DP matrix_AA3(A,n,n),matrix_A3(n,n),matrix_V(n,n);
Vec_DP matrix_W(n),matrix_B3(B,n),matrix_x3(n);
for(int i=0;i<n; i++){
    for(int j=0; j<n; j++){
        matrix_A3[i][j] = matrix_AA3[i][j];
    }
}

NR::svdcmp(matrix_A3, matrix_W, matrix_V);
NR::svbksb(matrix_A3, matrix_W, matrix_V, matrix_B3, matrix_x3);

cout << "solution by SVD\n";
for(int i=0; i<n; i++){
    cout << matrix_x3[i] << " ";
}
cout << "\n\n";
```

```
NR::gaussj(matrix_A,matrix_B);
for(int i=0; i<n; i++){
    cout << matrix_B[i][0] << " ";
}
cout << "\n\n";
```

```
NR::ludcmp(matrix_A, indx,d);
for(int i=0; i<n; i++){
    for(int j=0; j<n; j++){
        cout << matrix_A[i][j] << " ";
    }
    cout << "\n";
}
cout << "\n\n";
```

```
Mat_O_DP matrix_C(n,m);
Vec_O_DP diagonal_D(n);
NR::svdcmp(matrix_A, diagonal_D, matrix_C);
```

마지막으로 특이값 분해를 실행했다. 특이값 분해후 생기는 U, W, V를 이용하여 해를 구하고 출력해준다.

3. Result

3.1 "lineq1.dat"

```
solution by gauss-jordan elimination
1.66667 -1.66667 -0 -0.666667

inverse matrix of A
6.0048e+15 2.2518e+15 -2.2518e+15 -7.506e+14
1.20096e+16 4.5036e+15 -4.5036e+15 -1.5012e+15
-1.80144e+16 -6.7554e+15 6.7554e+15 2.2518e+15
-6.0048e+15 -2.2518e+15 2.2518e+15 7.506e+14

after LU decomposition
11 4 6 1
0.454545 1.18182 1.27273 -1.45455
-0.181818 -0.230769 -0.615385 1.84615
0.363636 0.461538 -0.375 5.55112e-16

d value :  1

solution by LU decomposition
1.9 -1.2 -0.7 -0.9

after iterative improvement
1.9 -1.2 -0.7 -0.9

solution by SVD
-0.858357 -6.71671 7.57507 1.85836
```

가우스 조르단을 통해 구한 해는 (5/3, -5/3, 0, -2/3)
LU분해를 통해 구한 해는 (1.9, -1.2, -0.7, -0.9)
특이값 분해를 통해 구한 해는 (-0.858357, -6.71671, 7.57507, 1.85836)
iterative improvement 이후 해는 (1.9, -1.2, -0.7, -0.9)
det(A) = 0, 역행렬은 없다.

3.2 "lineq2.dat“

```
solution by gauss-jordan elimination
-2.87357 -0.612357 0.976277 0.635819 -0.553441

inverse matrix of A
0.354536 0.766945 0.207769 -0.595412 0.253128
0.0354536 0.126694 0.195777 -0.159541 0.0503128
-0.138686 -0.0985401 -0.0967153 0.124088 0.0164234
-0.0521376 -0.303962 -0.0232013 0.234619 -0.0445777
0.149114 0.459333 0.0513556 -0.171011 0.0424922

after LU decomposition
5 0 8 7 -2
-0.2 6 1.6 4.4 1.6
0.4 -0.666667 -7.13333 5.13333 1.86667
0 0.166667 -0.383178 8.23364 5.4486
-0.2 0.166667 -0.46729 0.372304 2.17707

d value :  -1

solution by LU decomposition
-2.87357 -0.612357 0.976277 0.635819 -0.553441

after iterative improvement
-2.87357 -0.612357 0.976277 0.635819 -0.553441

solution by SVD
-2.87357 -0.612357 0.976277 0.635819 -0.553441
```

가우스 조르단을 통해 구한 해는
(-2.87357, -0.612357, 0.976277, 0.635819, -0.553441)
LU분해를 통해 구한 해는
(-2.87357, -0.612357, 0.976277, 0.635819, -0.553441)
특이값 분해를 통해 구한 해는
(-2.87357, -0.612357, 0.976277, 0.635819, -0.553441)
iterative improvement 이후 해는
(-2.87357, -0.612357, 0.976277, 0.635819, -0.553441)

det(A) = (-1) * 5 * 6 * -7.13333 * 8.23364 * 2.17707 = 3836

역행렬은 파란 동그라미 부분이다.

3.3 "lineq3.dat“

```
solution by gauss-jordan elimination
-0.326608 1.53229 -1.04482 -1.58745 2.92848 -2.21893

inverse matrix of A
-0.162205 0.122801 0.0240679 -0.0164308 -0.0228397 0.0461324
0.169407 -0.0411167 0.228313 -0.087624 0.180306 -0.395655
-0.0116364 0.122745 -0.117407 -0.180981 0.0159104 0.186766
0.105669 -0.0517256 -0.108916 0.299774 0.000858666 -0.190541
-0.053026 -0.0423615 0.160508 -0.224034 0.161811 0.0150242
-0.0623407 -0.0646943 -0.234216 0.351126 -0.364828 0.434633

after LU decomposition
7.8 8.3 7.7 3.3 1.9 4.8
0.0512821 7.77436 6.30513 1.73077 2.10256 5.05385
0.705128 0.379123 -4.8199 -1.9831 2.96313 1.09936
0.653846 -0.0420515 0.242635 4.19626 3.82715 1.70732
0.384615 0.271108 -0.192761 0.328693 4.81243 2.13444
0.448718 -0.131761 -0.638113 1.35403 1.19134 -2.74102

d value :  1

solution by LU decomposition
-0.326608 1.53229 -1.04482 -1.58745 2.92848 -2.21893

after iterative improvement
-0.326608 1.53229 -1.04482 -1.58745 2.92848 -2.21893

solution by SVD
-0.326608 1.53229 -1.04482 -1.58745 2.92848 -2.21893
```

가우스 조르단을 통해 구한 해는
(-0.326608, 1.53229, -1.04482, -1.58745, 2.92848, -2.21893)
LU분해를 통해 구한 해는
(-0.326608, 1.53229, -1.04482, -1.58745, 2.92848, -2.21893)
특이값 분해를 통해 구한 해는
(-0.326608, 1.53229, -1.04482, -1.58745, 2.92848, -2.21893)
iterative improvement 이후 해는
(-0.326608, 1.53229, -1.04482, -1.58745, 2.92848, -2.21893)
det(A) = (+1) * 7.8 * 7.77436 * -4.8199 * 4.19626 * 4.81243 * -2.74102
= 16178.4

역행렬은 파란 동그라미 부분이다.

4. Discussion

4.1 lineq1에서 해는 무한히 많다.

lineq1에서 LU분해를 통한 L, U 값은 다음과 같았다.

```
11 4 6 1
0.454545 1.18182 1.27273 -1.45455
-0.181818 -0.230769 -0.615385 1.84615
0.363636 0.461538 -0.375 5.55112e-16
```

이때 5.55112e-16은 다른 값들과 비교했을 때 매우 작은 값이므로 0임을 알 수 있고, A가 singular임을 알 수 있다. (det A = 0)
A = LU 꼴로 다시 써주면

$$\begin{bmatrix} 11 & 4 & 6 & 1 \\ 5 & 3 & 4 & -1 \\ -2 & -1 & -2 & 2 \\ 4 & 2 & 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.4545 & 1 & 0 & 0 \\ -0.1818 & -0.2307 & 1 & 0 \\ 0.3636 & 0.4615 & -0.375 & 1 \end{bmatrix} \times \begin{bmatrix} 11 & 4 & 6 & 1 \\ 0 & 1.1818 & 1.2727 & -1.4545 \\ 0 & 0 & -0.6153 & 1.8461 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Ax = LUx = b에서
Ux = c, Lc = b
Lc = b에서 c = (11, -1, -1.23077, 1e-6) ≈ (11, -1, -1.23077, 0)

Ux = c에서 x = (1-a, -3-2a, 2+3a, a)
a가 -2/3일때는 가우스 조르단의 해와, a가 1.85836일때는 SVD의 해와 같다는 것을 확인할 수 있다.

4.2 iterative improvement의 결과

iterative improvement는 해에 어떤 오차가 있음을 가정하고 해의 오차를 줄여나가는 과정이다. 이미 앞에 LU분해에서 완벽한 해를 찾았으므로 iterative improvement를 반복해도 해의 변화가 없는 것으로 예상된다.

5. Reference

[1] numerical analysis 10th edition
[2] http://numerical.recipes/book/book.html