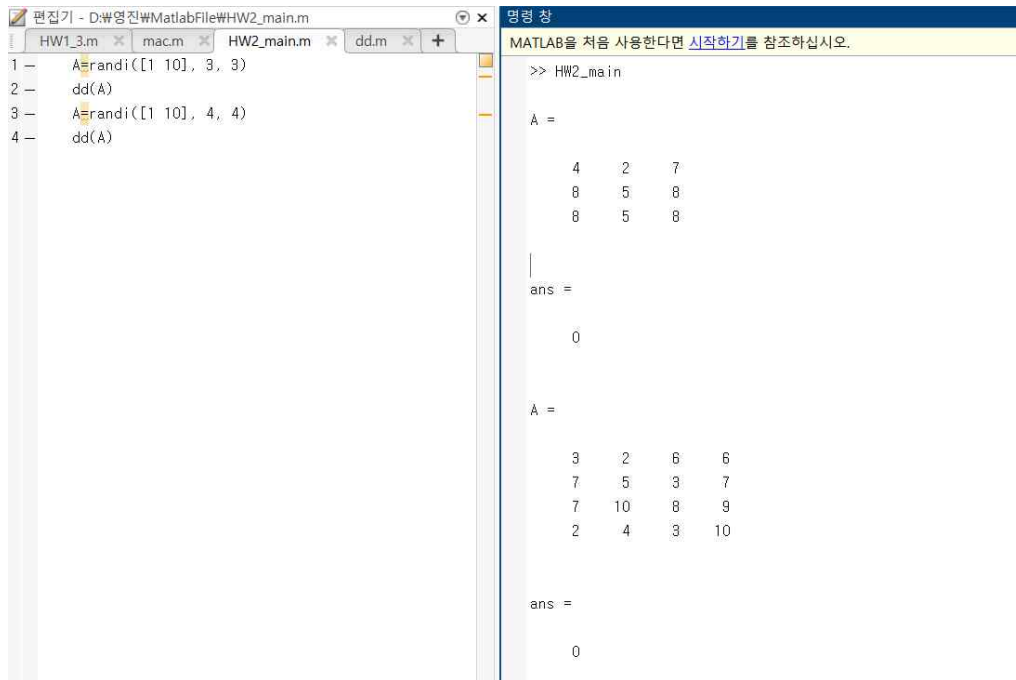


## 수치해석 HW2

201501489 최영진

### 1.

#### (1) 실행 결과



```
편집기 - D:\영진\MatlabFile\HW2_main.m
HW1_3.m  mac.m  HW2_main.m  dd.m  +
1 - A=randi([1 10], 3, 3);
2 - dd(A);
3 - A=randi([1 10], 4, 4);
4 - dd(A);

명령 창
MATLAB을 처음 사용한다면 시작하기를 참조하십시오.

>> HW2_main

A =

     4     2     7
     8     5     8
     8     5     8

ans =

     0

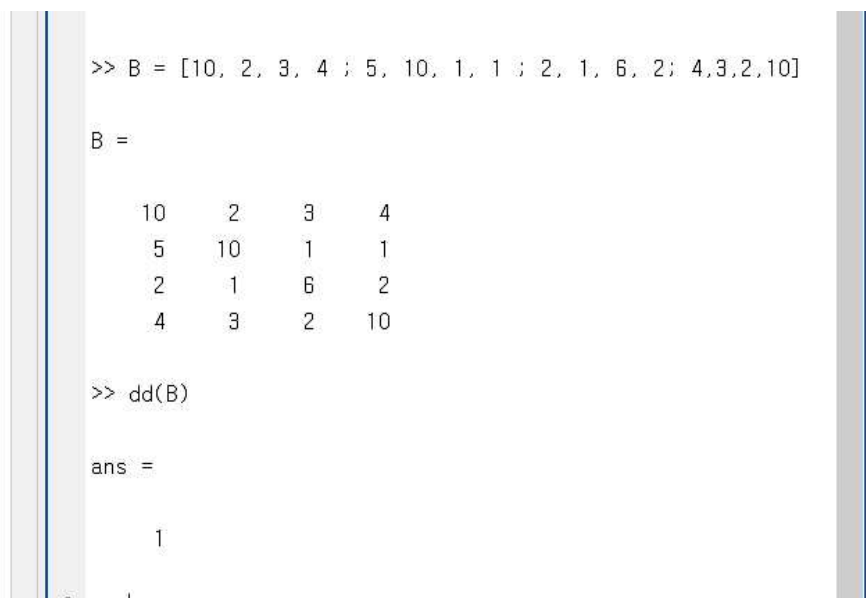
A =

     3     2     6     6
     7     5     3     7
     7    10     8     9
     2     4     3    10

ans =

     0
```

#### < main함수 실행 결과>



```
>> B = [10, 2, 3, 4 ; 5, 10, 1, 1 ; 2, 1, 6, 2 ; 4,3,2,10]

B =

    10     2     3     4
     5    10     1     1
     2     1     6     2
     4     3     2    10

>> dd(B)

ans =

     1
```

#### < 임의로 만든 강대각 지배 행렬 B를 넣은 결과>

(2) 코드 설명 (HW2\_main, dd.m)

```
function y=dd(A) %y = 1 or 0
    y = 1;
    for i = 1 : size(A,1)
        %dia는 diagonal
        dia = 0;
        %sum_r은 나머지 합
        sum_r = 0;
        for j = 1 : size(A,2)
            if (i == j)
                %dia는 diagonal
                dia = abs(A(i,j));
            else
                sum_r = sum_r + abs(A(i,j));
            end
        end
        if (dia <= sum_r)
            y = 0;
            break;
        end
    end
end
```

← for 문 2개를 중첩하여 들어온 행렬 A의 행과 열에 따라 코드 진행

← 각 i 행마다 대각의 절대값과 다른 계수 절대값의 합을 비교해 줄 dia와 sum\_r

← 만약 i와 j가 같다면 dia에 절대값을 넣어줌

← i와 j가 다르다면 절대값을 합해줌

← 강대각 지배 행렬이 아니라면 0 출력, 더 이상 확인할 필요가 없으므로 for문 탈출

(3) 토의 사항

코드가 깔끔하게 작성된 것 같지는 않아 아쉽다. 내용 면으로는 메인에서 제작되는 행렬이 랜덤이라 강대각 지배 행렬이 한번에 나오기는 어려워서 따로 제대로 실행하는지 확인해보고자 강대각 지배 행렬 B를 만들어 함수에 넣어 돌려보았다. 결과는 예상대로 1을 출력해냈고 dd함수가 제대로 동작함을 확인할 수 있었다.

## 2.

### (1) 실행 결과

```
1 %2번 문제
2 %Ax = B
3 clear; clc;
4 A = [ 1 2 2 ; 4 4 2 ; 4 6 4 ]
5 B = [ 3 ; 6 ; 10 ]
6 answer = A*B
7
8 disp("부분 피벗팅을 시작합니다.")
9 n = length(B);
10
11 for i = 1 : n-1
12     %A에 마이너스 원소가 존재하는 것 포함
13     AC = abs(A);
14     %i열의 최대값 MX
15     [MX, I] = max(AC(i:end, i));
16     I = I + i-1; %최대값이 존재하는 위치
17     fprintf('%d열의 최대값은 %d입니다.\n', i, MX);
18     if (I ~= i)
19         %A자리바꿈
20         temp = A(i,:);
21         A(i,:) = A(I,:);
22         A(I,:) = temp;
23         %B자리 바꿈
24         temp_1 = B(i);
25         B(i) = B(I);
26         B(I) = temp_1;
27     end
28
29 %이 스코프를 빠져나갈 때 까지
```

명령 창

MATLAB을 처음 사용한다면 [시작하기](#)를 참조하십시오.

answer =

-1  
3  
-1

부분 피벗팅을 시작합니다.  
1열의 최대값은 4입니다.  
2열의 최대값은 2입니다.

A =

4.0000 4.0000 2.0000  
0 2.0000 2.0000  
0 0 0.5000

B =

6.0000  
4.0000  
-0.5000

x =

-1 3 -1

<전체 실행 화면>

1열의 최대값은 4입니다.  
2열의 최대값은 2입니다.

<마지막 열을 제외한 각 열의 최대값 출력 화면>

### (2) 코드 설명 (HW2\_2.m)

우선 정답을 A\*B로 추출해내고 시작했다. 함수의 계수가 '1'일수도 있으므로 각 열(마지막 열 제외)의 최대값을 구하는 것은 A행렬에 절대값을 씌운 AC행렬로 진행한다.  $[MX, I] = \max(AC(i:end, i))$ ; 는  $(n-i+1, n)$ 행렬이므로 행의 위치를 알려주는 I값에는 i-1만큼을 더하여 실제 위치로 보정한다. 그 뒤 자리 바꿈과 위삼각행렬화는 보이는 그대로이다. 최대값을 가진 행을 위로 올리고, 그 아래 행의 불필요한 계수들을 0으로 만들기 위해 계산해준다.

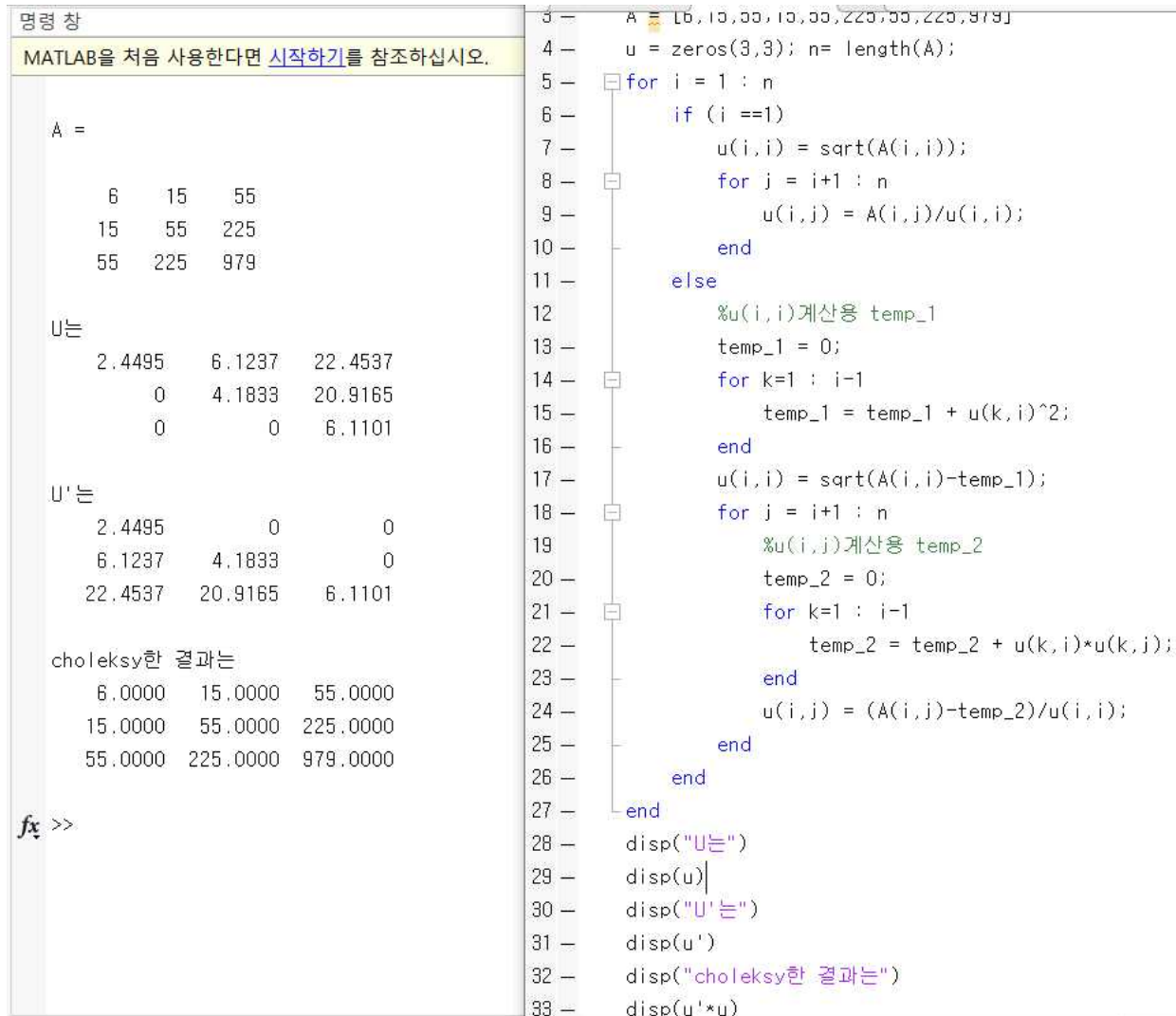
### (3) 토의 사항

위삼각행렬화를 할 때, 초반에 B값이 아닌 A값을 먼저 계산했다가 이상한 결과값을 출력했었다. A행렬의 값이 아직 바뀌지 않은 상태에서 B값을 먼저 바꿔야 함을 유념해야겠다.

확인 과정에서는 정답을 비교하는 것 외에 직접 손으로 풀어가며 각 단계별 확인도 진행했는데, 나누는 계수가 작지 않아서 수월하게 진행할 수 있었다.

### 3.

#### (1) 실행 결과



The image shows a MATLAB Command Window on the left and a script editor on the right. The Command Window displays the results of the script, including the matrix A, the upper triangular matrix U, the lower triangular matrix U', and the Cholesky decomposition result. The script editor shows the code used to calculate these values.

```
명령 창
MATLAB을 처음 사용한다면 시작하기를 참조하십시오.

A =

     6     15     55
    15     55    225
    55    225    979

U는
    2.4495    6.1237   22.4537
         0    4.1833   20.9165
         0         0    6.1101

U'는
    2.4495         0         0
    6.1237    4.1833         0
   22.4537   20.9165    6.1101

choleksy한 결과는
    6.0000   15.0000   55.0000
   15.0000   55.0000  225.0000
   55.0000  225.0000  979.0000

fx >>

3 = A = [6, 15, 55, 15, 55, 225, 55, 225, 979]
4 = u = zeros(3,3); n= length(A);
5 = for i = 1 : n
6 =     if (i ==1)
7 =         u(i,i) = sqrt(A(i,i));
8 =     for j = i+1 : n
9 =         u(i,j) = A(i,j)/u(i,i);
10 =     end
11 =     else
12 =         %u(i,i)계산용 temp_1
13 =         temp_1 = 0;
14 =         for k=1 : i-1
15 =             temp_1 = temp_1 + u(k,i)^2;
16 =         end
17 =         u(i,i) = sqrt(A(i,i)-temp_1);
18 =         for j = i+1 : n
19 =             %u(i,j)계산용 temp_2
20 =             temp_2 = 0;
21 =             for k=1 : i-1
22 =                 temp_2 = temp_2 + u(k,i)*u(k,j);
23 =             end
24 =             u(i,j) = (A(i,j)-temp_2)/u(i,i);
25 =         end
26 =     end
27 = end
28 = disp("U는")
29 = disp(u)
30 = disp("U'는")
31 = disp(u')
32 = disp("choleksy한 결과는")
33 = disp(u'*u)
```

#### (2) 코드 설명 (HW2\_3)

i가 1일 때는 별다른 추가 계산없이  $a(i,i)$ 와  $a(i,j)$ ,  $u(i,i)$ 만을 사용하면 되기에 조건문을 열어 처리했다. 그 외에는  $\Sigma$  계산이 필요하기에 for문을 열어 각각 sigma를 계산해 담아둘 temp변수들을 설정해두었다.

#### (3) 토의 사항

주어진 수식을 그대로 코드로 옮겨 작성하고 뽑아낸 위삼각행렬 U와 그 전치행렬 U'(아래삼각행렬)을 곱해서(U' U) 기존 행렬 A값과 일치하는 값이 나오는지 확인하는 방식으로 진행했다.  $u(i,i)$ 를 결정하는 코드의 위치를  $u(i,j)$  결정 코드의 바로 위에 두었다가 마지막 n행n열이 설정되지 않는 실수도 있었으나 위치를 바꾸고 나서는 기존 A행렬의 형태와 동일한 값이 나옴을 확인할 수 있었다.

#### 4.

##### (1) 실행 결과

```
명령 창
MATLAB을 처음 사용한다면 시작하기를 참조하십시오.

A =
     6     15     55
    15     55    225
    55    225    979

B =
     1
     2
     3

y =
    0.4082   -0.1195   -0.6001

x =
   -0.0893    0.4625   -0.0982
```

```
>> A\B

ans =

   -0.0893
    0.4625
   -0.0982
```

<나와야 하는 답>

<실행결과>

##### (2) 코드 설명 (HW2\_4.m)

문제 3에서 작성한 코드를 그대로 복사해 옮긴 후  $U'y = B$ ,  $Ux = y$  순으로 전진대입법과 역진대입법을 수행했다.

1step :  $Ly = B$  ( $L = U'$ ) → 전진대입법은  $y(1)$ 을 구하고,  $y(1)$ 을 그 계수만큼 곱해서 제거한 후  $y(2)$ 의 계수만큼 나눠서  $y(2)$ 를 구하는 형식을 취한다. 즉,  $y(n)$ 을 구하는 각 단계에서 해당 식에 존재하는 모든  $y(k)$  ( $k = 1, 2, \dots, n-1$ )들이 그 수만큼 제거된 후  $y(n)$ 에 곱해진 계수로 나누면 된다.

2step :  $Ux = y$  → 역진대입법은 1step에서 구한 것들이 정확하게 반대되는 순서로 구하는 방법이다.  $x(n)$ 이 가장 먼저 구해지므로 그다음은  $x(n-1)$ 을 구하고 마지막으로  $x(1)$ 을 구한다. 구하려는  $x$ 값 외의 다른  $x$ 값들은 모두 빼주는 것은 동일하다.

##### (3) 토의 사항

계산 시에 정확한 행렬 기호를 제대로 넣어야 하는 것만 주의하면 그 외에 어려운 사항은 없다.