

Numerical Linear Algebra

Programming Assignment #01

2015-17231

박우정

Exercise 2.6.

MATLAB으로 $A = LU$ factorization without pivoting을 구현한 코드는 아래와 같다.

```
ALU2_6
A=input('LU분해를 알고 싶은 정사각행렬을 입력하십시오: ');
[n,m]=size(A);
if (n~=m)
    disp('정사각행렬이 아닙니다. ');
    return;
end;
saveA=A;
L=eye(n);
U=zeros(n,n);
for i=1:n
    L(i+1:n,i)=A(i+1:n,i)/A(i,i);
    for j=i+1:n
        A(j,:)=A(j,:)-L(j,i)*A(i,:);
    end
end
U=A;
disp('입력받은 행렬: ')
disp(saveA)
disp('상삼각행렬: ')
disp(U)
disp('하삼각행렬: ')
disp(L)
for i=1:n
    for j=1:n
        A(i,j)=dot(L(i,:),U(:,j));
    end
end
if (saveA==A)
    disp("L 곱하기 U=")
    A
    disp("같습니다.")
else
    disp("같지 않습니다. 다시 짜야 할 것 같습니다.")
end
```

위 스크립트는 정사각행렬을 받으면 입력받은 행렬을 출력한 뒤, Gauss Elimination을 통해 LU factorization을 시행하여 L 과 U 를 출력해준다. 이후에 L 과 U 를 곱하면 처음에 입력받은 행렬 A 가 나오는지 확인해준다.

먼저, (2.2)에서 주어진 행렬 $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$ 의 LU factorization을 시행해보자. MATLAB

문법에 맞추어 [2 -1 0;-1 2 -1; 0 -1 2]를 input하면 실행 결과는 아래와 같다.

```
>> ALU2_6
LU분해를 알고 싶은 정사각행렬을 입력하시오: [2 -1 0;-1 2 -1; 0 -1 2]
입력받은 행렬:
    2    -1     0
   -1     2    -1
    0    -1     2

상삼각행렬:
    2.0000   -1.0000     0
         0    1.5000   -1.0000
         0         0    1.3333

하삼각행렬:
    1.0000         0         0
   -0.5000    1.0000         0
         0   -0.6667    1.0000

L 곱하기 U=

A =

    2    -1     0
   -1     2    -1
    0    -1     2

같습니다.
```

처음 입력받은 행렬과 LU factorization을 시행한 L , U 의 행렬곱이 같다고 출력되었으므로, 알맞게 프로그래밍되었다.

다른 예시로, 다음과 같은 3×3 행렬 $B = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}$ 를 LU factorization을 시행해보자.

MATLAB 문법에 맞추어 [2 1 1;4 -6 0;-2 7 2]를 input하면 실행 결과는 아래와 같다.

```

>> ALU2_6
LU분해를 알고 싶은 정사각행렬을 입력하시오: [2 1 1;-6 0;-2 7 2]
입력받은 행렬:
    2    1    1
    4   -6    0
   -2    7    2

상삼각행렬:
    2    1    1
    0   -8   -2
    0    0    1

하삼각행렬:
    1    0    0
    2    1    0
   -1   -1    1

L 곱하기 U=

A =

    2    1    1
    4   -6    0
   -2    7    2

같습니다.

```

프로그래밍의 결과로 '같습니다'를 얻었고, 실제로 계산해보아도 위와 같은 결과를 얻는다. 따라서 알맞게 프로그래밍되었다고 할 수 있다.

Exercise 2.7.(with analytic hand out)

F90 program 2.1을 참조하여 MATLAB code로 옮긴 것이 아래와 같다.

Check2_7

```
n=input('What is the dimension n? ');
n=n-1;
A=zeros(n,n);
b=zeros(n,1);
saveb=zeros(n,1);
saveA=zeros(n,n);
for j=1:n
    A(j,j)=2;
end
for j=1:n-1
    A(j,j+1)=-1;
    A(j+1,j)=-1;
end
b(1)=1;
saveA=A;
saveb=b;
for j=1:n-1
    for k=j+1:n
        m=A(k,j)/A(j,j);
        A(k,j:n)=A(k,j:n)-m*A(j,j:n);
        b(k)=b(k)-m*b(j);
    end
end
for j=n:-1:1
    b(j)=(b(j)-dot(A(j,j+1:n),b(j+1:n)))/A(j,j);
end
for j=1:n
    if(abs(saveb(j)-dot(saveA(j,1:n),b(1:n)))>5*10^-13)
        fprintf('The computed solution seems to be wrong at %f \n',j);
    end
end
disp('The solution x is as follows: ')
disp(b)
```

이제 각 $N = 10, 20, 100$ 일 때를 실행해보면 다음과 같은 결과를 얻을 수 있다.

```
>> Check2_7
What is the dimension
n? 10
The solution x is as
follows:
0.9000
0.8000
0.7000
0.6000
0.5000
0.4000
0.3000
0.2000
0.1000
```

```
>> Check2_7
What is the dimension
n? 20
The solution x is as
follows:
0.9500
0.9000
0.8500
0.8000
0.7500
0.7000
0.6500
0.6000
0.5500
0.5000
0.4500
0.4000
0.3500
0.3000
0.2500
0.2000
0.1500
0.1000
0.0500
```

```
>> Check2_7
What is the dimension n? 100
The solution x is as follows:
0.9900
0.9800
0.9700
0.9600
0.9500
0.9400
0.9300
0.9200
0.9100
0.9000
0.8900
0.8800
0.8700
0.8600
0.8500
0.8400
0.8300
0.8200
0.8100
0.8000
0.7900
0.7800
0.7700
0.7600
0.7500
0.7400
0.7300
0.7200
0.7100
0.7000
0.6900
0.6800
0.6700
0.6600
0.6500
0.6400
0.6300
0.6200
0.6100
0.6000
0.5900
0.5800
0.5700
0.5600
0.5500
0.5400
0.5300
0.5200
0.5100
0.5000
0.4900
0.4800
0.4700
0.4600
0.4500
0.4400
0.4300
0.4200
0.4100
0.4000
0.3900
0.3800
0.3700
0.3600
0.3500
0.3400
0.3300
0.3200
0.3100
0.3000
0.2900
0.2800
0.2700
0.2600
0.2500
0.2400
0.2300
0.2200
0.2100
0.2000
0.1900
0.1800
0.1700
0.1600
0.1500
0.1400
0.1300
0.1200
0.1100
0.1000
0.0900
0.0800
0.0700
0.0600
0.0500
0.0400
0.0300
0.0200
0.0100
```