2. THE MOORE-PENROSE PSEUDO-INVERSE EXERCISE

Creating Data.

A) Creating a random Matrix D of size mx4.

```
m=4
```

```
m = 4
```

```
D =randi([1,5],[m,4])
```

```
D = 4×4
     4     4     4     5
     3     3     1     2
     1     1     4     4
     1     2     2     1
```

```
D(:,4) = 4*D(:,1) -3*D(:,2) +2*D(:,3) -1
```

```
D = 4×4
     4     4     4    11
     3     3     1     4
     1     1     4     8
     1     2     2     1
```

B) Introducing Small additive errors in data.

```
eps = 1.e-4
```

```
eps = 1.0000e-04
```

```
D=D*eps
```

```
D = 4×4
    0.0004    0.0004    0.0004    0.0011
    0.0003    0.0003    0.0001    0.0004
    0.0001    0.0001    0.0004    0.0008
    0.0001    0.0002    0.0002    0.0001
```

2. Find the coefficients of x solving Dx = b.

A) Compute the Singular Value Decomposition.

```
[U,S,V]=svd(D)
```

```
U = 4×4
   -0.7757   -0.1305   -0.2303   -0.5729
   -0.3258   -0.6454   -0.1897    0.6643
   -0.5211    0.6842    0.2042    0.4675
   -0.1437   -0.3134    0.9323   -0.1087
S = 4×4
    0.0017         0         0         0
         0    0.0004         0         0
         0         0    0.0002         0
         0         0         0    0.0000
```

```
V = 4×4
   -0.2834   -0.5948   -0.1903    0.7278
   -0.2920   -0.6841    0.3114   -0.5914
   -0.3465    0.2686    0.8453    0.3055
   -0.8452    0.3257   -0.3903   -0.1650
```

B) Compute Moore-Penrose pseudo-inverse using A) and find x

```
for i =1:1:3
    if(S(i,i)>0)
        S(i,i)=1/S(i,i);
    end
end
S=S.'
```

```
S = 4×4
10^3 ×
    0.5974         0         0         0
         0    2.8493         0         0
         0         0    5.3807         0
         0         0         0    0.0000
```

```
A_dagger=V*S*U.'
```

```
A_dagger = 4×4
10^3 ×
    0.5882    1.3432   -1.2805   -0.3989
    0.0039    0.9970   -0.9006    2.1983
   -0.9866   -1.2893    1.5604    4.0303
    0.7541   -0.0361    0.4692   -2.1765
```

```
b=[1;1;1;1]
```

```
b = 4×1
     1
     1
     1
     1
```

```
x1=A_dagger*b
```

```
x1 = 4×1
10^3 ×
    0.2520
    2.2986
    3.3148
   -0.9893
```

C) Compute using Matlab built-in function

```
A_Dagger_Matlab=pinv(D)
```

```
A_Dagger_Matlab = 4×4
10^4 ×
   -5.0000    6.0000    4.0000   -1.0000
    4.1111   -4.6667   -3.4444    1.0000
   -2.2222    2.3333    1.8889   -0.0000
    1.2222   -1.3333   -0.8889    0.0000
```

```
x2=A_Dagger_Matlab*b
```

```
x2 = 4×1
10^4 ×
    4.0000
   -3.0000
    2.0000
   -1.0000
```

```
m1=3
```

```
m1 = 3
```

```
D2 =randi([1,5],[m1,4])
```

```
D2 = 3×4
     2     4     4     4
     1     3     4     5
     3     3     4     2
```

```
D2(:,4) = 4*D2(:,1) -3*D2(:,2) +2*D2(:,3) -1
```

```
D2 = 3×4
     2     4     4     3
     1     3     4     2
     3     3     4    10
```

```
eps = 1.e-4
```

```
eps = 1.0000e-04
```

```
D2=D2*eps
```

```
D2 = 3×4
10^-3 ×
    0.2000    0.4000    0.4000    0.3000
    0.1000    0.3000    0.4000    0.2000
    0.3000    0.3000    0.4000    1.0000
```

```
[U2,S2,V2]=svd(D2)
```

```
U2 = 3×3
   -0.4508    0.5737    0.6838
   -0.3490    0.5919   -0.7266
   -0.8216   -0.5662   -0.0666
S2 = 3×4
    0.0014         0         0         0
         0    0.0004         0         0
         0         0    0.0001         0
V2 = 4×4
   -0.2699    0.0093    0.6136   -0.7420
   -0.3861    0.5445    0.4948    0.5565
   -0.4711    0.5503   -0.6081   -0.3246
   -0.7458   -0.6329   -0.0941    0.1855
```

```
for i =1:1:3
    if(S2(i,i)>0)
        S2(i,i)=1/S2(i,i);
    end
end
```

```
S2=S2.'
```

```
S2 = 4×3
10^4 ×
    0.0726         0         0
         0    0.2296         0
         0         0    1.3905
         0         0         0
```

```
A_D=V2*S2*U2.'
```

```
A_D = 4×3
10^3 ×
    5.9355   -6.1183   -0.4194
    5.5484   -4.1613   -0.9355
   -4.9032    7.0108    0.1290
   -1.4839    0.2796    1.3548
```

```
b2=[1;1;1]
```

```
b2 = 3×1
     1
     1
     1
```

```
X2=A_D*b2 % VALUE OF X solved manually
```

```
X2 = 4×1
10^3 ×
   -0.6022
    0.4516
    2.2366
    0.1505
```

```
A_D_M=pinv(D2)
```

```
A_D_M = 4×3
10^3 ×
    5.9355   -6.1183   -0.4194
    5.5484   -4.1613   -0.9355
   -4.9032    7.0108    0.1290
   -1.4839    0.2796    1.3548
```

```
X2_M=A_D_M*b2 % VALUE OF X on applying pinv()
```

```
X2_M = 4×1
10^3 ×
   -0.6022
    0.4516
    2.2366
    0.1505
```

```
null(D2)
```

```
ans = 4×1
   -0.7420
    0.5565
   -0.3246
```

```
0.1855
```

Final Outcome := Since m=3 we have infinitely many solutions thus we get a non-empty null vector as above. The set of all possible solutions are X2_M + lamda*null(D2).