Obtaining the basis for rowspace, columnspace, nullspace and the nullity of a given matrix

```
A=randi([0,9],5,2)*randi([0,9],2,5)
```

```
A = 5×5
    95    36    44    44    26
   153    54    72    72    45
   135    46    64    64    41
    48    12    24    24    18
   127    44    60    60    38
```

```
[RR, ic]=rref(A);
r = length(ic);
R=RR(1:r,:)
```

```
R = 2×5
    1.0000         0    0.5714    0.5714    0.5714
         0    1.0000   -0.2857   -0.2857   -0.7857
```

```
C=A(:,ic)
```

```
C = 5×2
    95    36
   153    54
   135    46
    48    12
   127    44
```

```
N=null(A)
```

```
N = 5×3
    0.6252   -0.0105    0.0000
   -0.4179    0.3850   -0.0000
   -0.4417   -0.3706   -0.7071
   -0.4417   -0.3706    0.7071
   -0.2106    0.7596   -0.0000
```

```
S=size(N);
Nullity=S(:,2)
```

```
Nullity =
3
```

3D scatterplot of Row space and column space of any random × matrix with rank 2.

```
A = randi([-3, 3], 3, 2) * randi([-3, 3], 2, 3)
```

```
A = 3×3
    -2     1    -2
    -6    -7    -6
     4     6     4
```

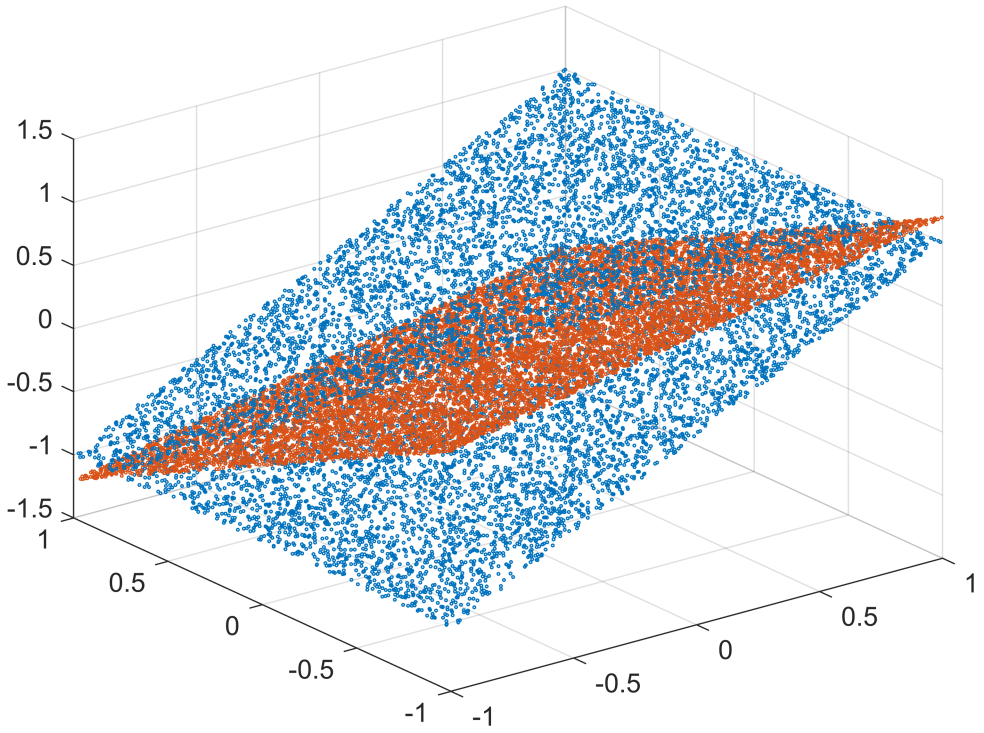
```
[RR, ic] = rref(A);
r = length(ic);
R = RR(1:r, :)
```

```
R = 2×3
     1     0     1
     0     1     0
```

```
RSB1 = R(1, :)';
RSB2 = R(2, :)';
[RR, ic] = rref(A');
r = length(ic);
C = RR(1:r, :)
```

```
C = 2×3
    1.0000         0    0.4000
         0    1.0000   -0.8000
```

```
CSB1 = C(1, :)';
CSB2 = C(2, :)';
RSpts = [];
CSpts = [];
for i = 1:10000
k1 = -1 + 2 * rand(1);
k2 = -1 + 2 * rand(1);
a1 = -1 + 2 * rand(1);
a2 = -1 + 2 * rand(1);
RSpts = [RSpts, k1 * RSB1 + k2 * RSB2];
CSpts = [CSpts, a1 * CSB1 + a2 * CSB2];
end
scatter3(RSpts(1, :), RSpts(2, :), RSpts(3, :), 1);
hold on
scatter3(CSpts(1, :), CSpts(2, :), CSpts(3, :), 1);
hold off
```



1

```
M=[1 0 0;0 0 0;0 0 3]
```

```
M = 3×3
     1     0     0
     0     0     0
     0     0     3
```

```
[RR, ic] = rref(M);
r = length(ic);
R = RR(1:r, :)
```

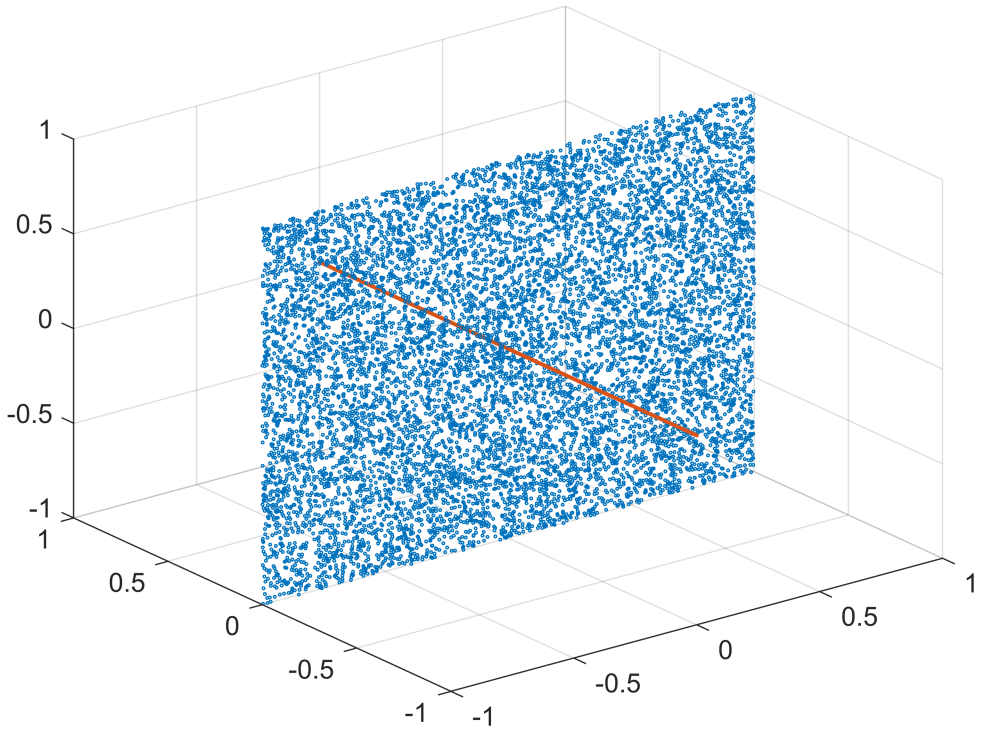
```
R = 2×3
     1     0     0
     0     0     1
```

```
N=null(M)
```

```
N = 3×1
     0
     1
     0
```

```
RSB1 = R(1, :)';
RSB2 = R(2, :)';
RSpts = [];
NSpts=[];
for i = 1:10000
```

```
k1 = -1 + 2 * rand(1);
k2 = -1 + 2 * rand(1);
a1=-1+2*rand(1);
RSpts = [RSpts, k1 * RSB1 + k2 * RSB2];
NSpts=[NSpts,a1*N];
end
scatter3(RSpts(1, :), RSpts(2, :), RSpts(3, :), 1);
hold on
scatter3(NSpts(1, :), NSpts(2, :), NSpts(3, :), 1);
hold off
```



2a

```
A=[1 4;0 5]
```

```
A = 2×2
     1     4
     0     5
```

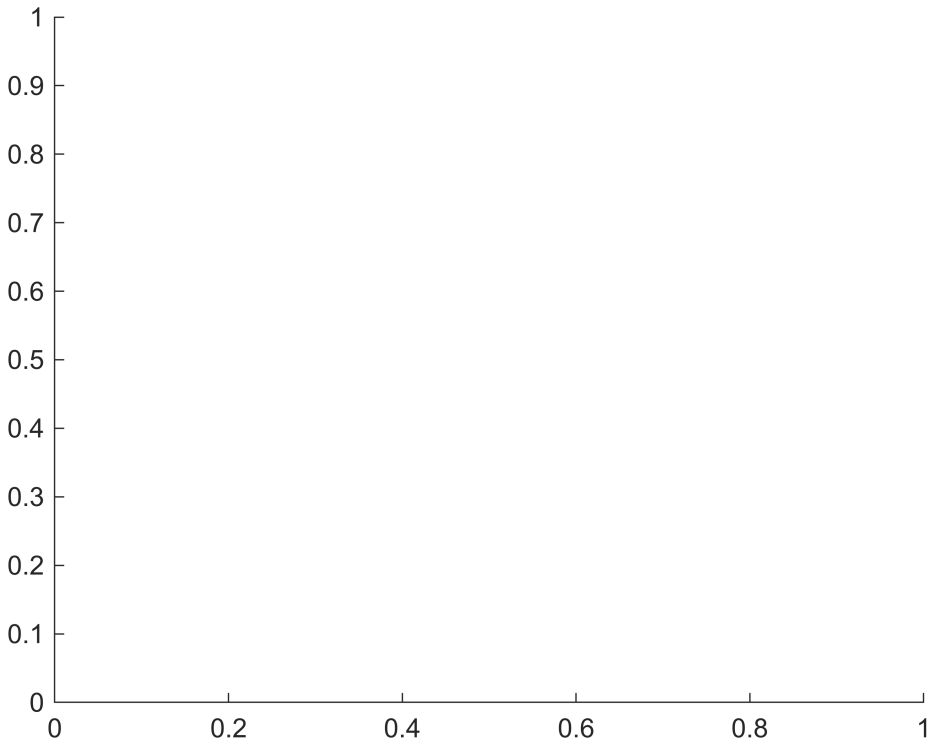
```
N=null(A)
```

```
N =

  2×0 empty double matrix
```

```
pts=[];
for i=1:1000
k1=-1+2*rand(1);
```

```
pts=[pts,k1*N];
end
scatter(pts(1,:),pts(2,:),1);
```



2b

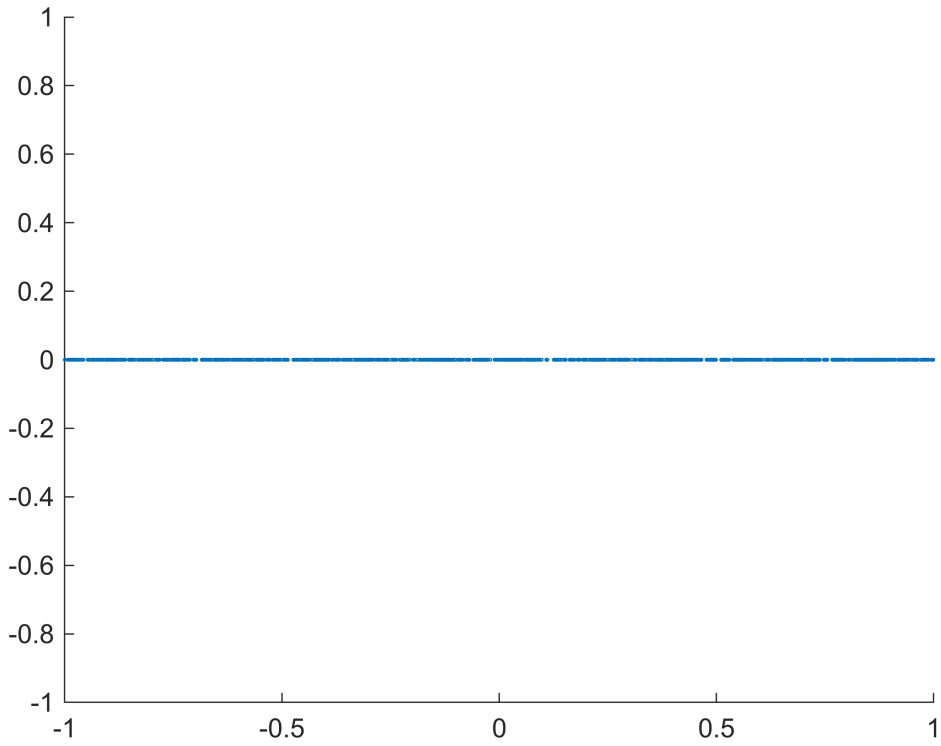
```
B=[0 0;0 5]
```

```
B = 2×2
     0     0
     0     5
```

```
N=null(B)
```

```
N = 2×1
     1
     0
```

```
pts=[];
for i=1:1000
k1=-1+2*rand(1);
pts=[pts,k1*N];
end
scatter(pts(1,:),pts(2,:),1);
```



2c

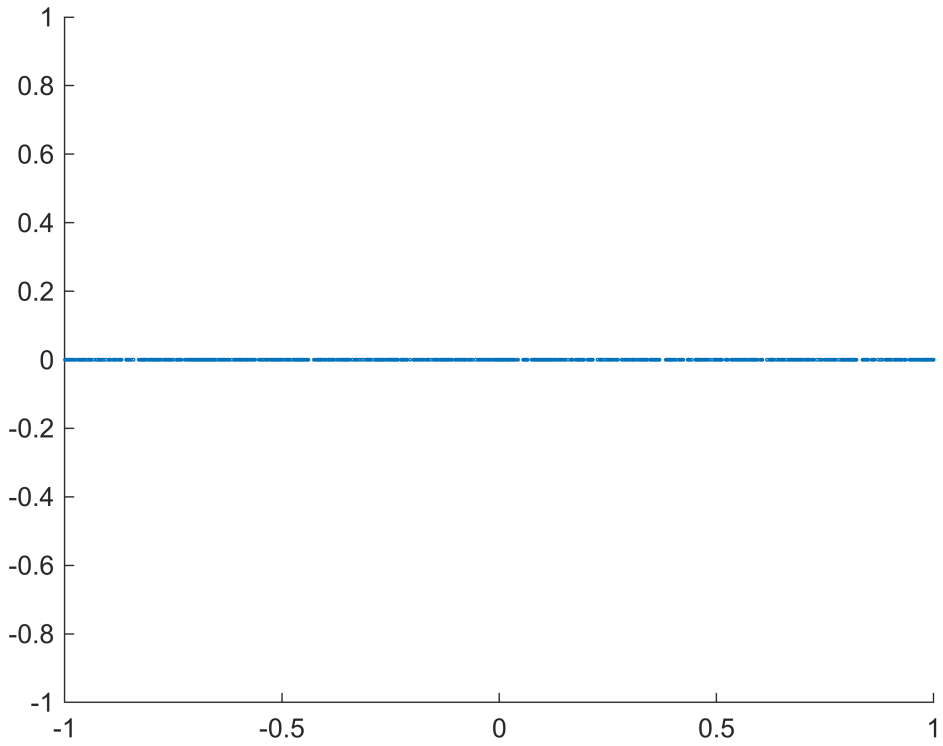
```
C=[0 0;0 0]
```

```
C = 2×2
     0     0
     0     0
```

```
N=null(C)
```

```
N = 2×2
     1     0
     0     1
```

```
NSB1 = R(1, :)';
NSB2 = R(2, :)';
pts=[];
for i=1:1000
k1=-1+2*rand(1);
k2=-1+2*rand(1);
pts=[pts,k1*NSB1+k2*NSB2];
end
scatter(pts(1,:),pts(2,:),1);
```



3

```
A = randi([-3, 3], 3, 2) * randi([-3, 3], 2, 3)
```

```
A = 3×3
     0     4    -1
    -7    -5    -4
     1    -1     1
```

```
[RR, ic] = rref(A);
r = length(ic);
R = RR(1:r, :)
```

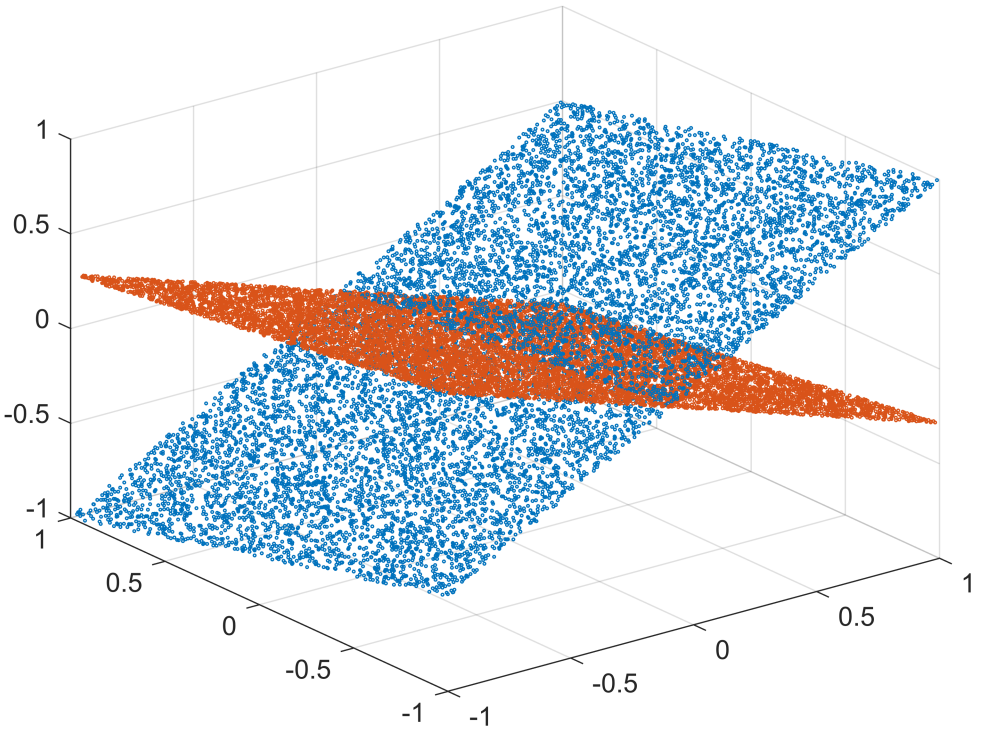
```
R = 2×3
    1.0000         0    0.7500
         0    1.0000   -0.2500
```

```
RSB1 = R(1, :)';
RSB2 = R(2, :)';
[RR, ic] = rref(A');
r = length(ic);
C = RR(1:r, :)
```

```
C = 2×3
    1.0000         0   -0.4286
         0    1.0000   -0.1429
```

```
CSB1 = C(1, :)';
CSB2 = C(2, :)';
```

```
RSpts = [];
CSpts = [];
for i = 1:10000
k1 = -1 + 2 * rand(1);
k2 = -1 + 2 * rand(1);
a1 = -1 + 2 * rand(1);
a2 = -1 + 2 * rand(1);
RSpts = [RSpts, k1 * RSB1 + k2 * RSB2];
CSpts = [CSpts, a1 * CSB1 + a2 * CSB2];
end
scatter3(RSpts(1, :), RSpts(2, :), RSpts(3, :), 1);
hold on
scatter3(CSpts(1, :), CSpts(2, :), CSpts(3, :), 1);
hold off
```



4

```
M = randi([-3, 3], 3, 1) * randi([-3, 3], 1, 3)
```

```
M = 3×3
    -3    -2    -1
     6     4     2
     0     0     0
```

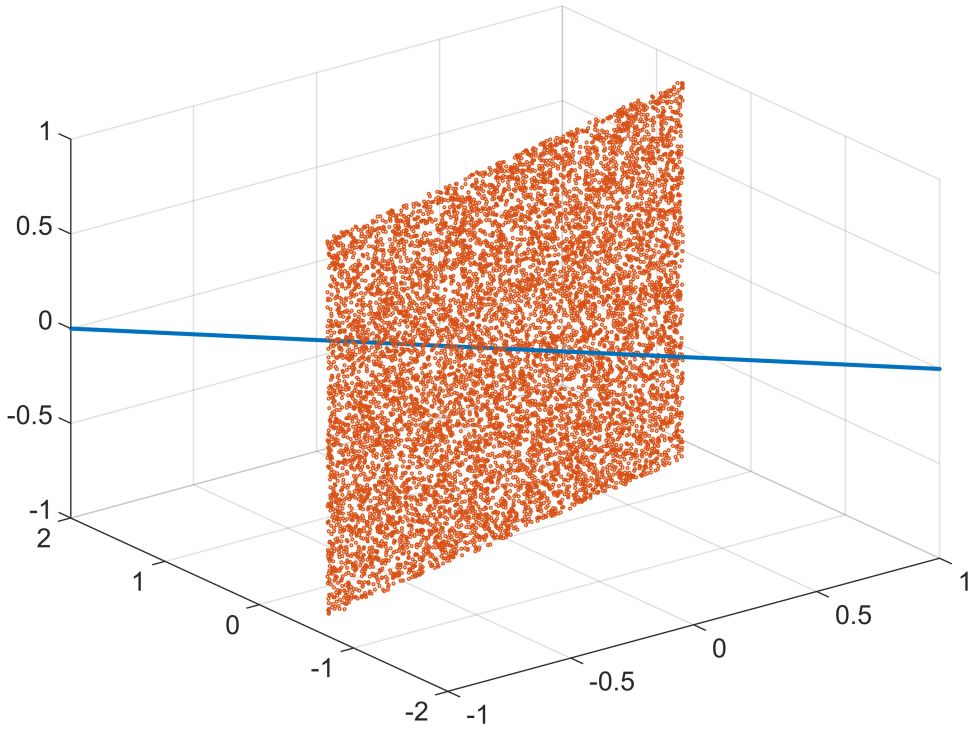
```
[RR, ic] = rref(M');
r = length(ic);
C = RR(1:r, :)
```

```
C = 1×3
     1    -2     0
```

```
LN=null(M')
```

```
LN = 3×2
    0.8944         0
    0.4472         0
         0    1.0000
```

```
CSB1 = C(1, :)';
LNSB1=LN(:,1);
LNSB2=LN(:,2);
CSpts = [];
LNSpts=[];
for i = 1:10000
k1 = -1 + 2 * rand(1);
a2 = -1 + 2 * rand(1);
a1=-1+2*rand(1);
CSpts = [CSpts, k1 * CSB1 ];
LNSpts=[LNSpts,a1*LNSB1+a2*LNSB2];
end
scatter3(CSpts(1, :), CSpts(2, :), CSpts(3, :), 1);
hold on
scatter3(LNSpts(1, :), LNSpts(2, :), LNSpts(3, :), 1);
hold off
```



5a

```
A=[1 3 4 7;2 4 6 10;3 5 8 13;4 6 10 16]
```

```
A = 4×4
     1     3     4     7
     2     4     6    10
     3     5     8    13
     4     6    10    16
```

```
u=[-2;-3;1;1]
```

```
u = 4×1
    -2
    -3
     1
     1
```

```
if rank([A u])==rank(A)
disp('u is a column space of A')
elseif rank ([A; u'])==rank(A)
disp('u is a row space of A')
elseif A*u == 0
disp('u is in the null space of A')
elseif transpose(u)*A == 0
disp('u is in the left null space of A')
else
disp('u is not in any space of A')
end
```

```
u is in the null space of A
```

```

```

5b

```
A=[1 3 4 7;2 4 6 10;3 5 8 13;4 6 10 16]
```

```
A = 4×4
     1     3     4     7
     2     4     6    10
     3     5     8    13
     4     6    10    16
```

```
v=[5;8;11;14]
```

```
v = 4×1
     5
     8
    11
    14
```

```
if rank([A v])==rank(A)
disp('v is a column space of A')
elseif rank ([A; v'])==rank(A)
disp('v is a row space of A')
```

```
elseif A*v == 0
disp('v is in the null space of A')
elseif transpose(v)*A == 0
disp('v is in the left null space of A')
else
disp('v is not in any space of A')
end
```

```
v is a column space of A
```

5c

```
A=[1 3 4 7;2 4 6 10;3 5 8 13;4 6 10 16]
```

```
A = 4×4
     1     3     4     7
     2     4     6    10
     3     5     8    13
     4     6    10    16
```

```
w=[1;1;2;3]
```

```
w = 4×1
     1
     1
     2
     3
```

```
if rank([A w])==rank(A)
disp('w is a column space of A')
elseif rank ([A; w'])==rank(A)
disp('w is a row space of A')
elseif A*w == 0
disp('w is in the null space of A')
elseif transpose(w)*A == 0
disp('w is in the left null space of A')
else
disp('w is not in any space of A')
end
```

```
w is a row space of A
```

5d

```
A=[1 3 4 7;2 4 6 10;3 5 8 13;4 6 10 16]
```

```
A = 4×4
     1     3     4     7
     2     4     6    10
     3     5     8    13
     4     6    10    16
```

```
y=[1;2;0;-1]
```

```
y = 4×1
     1
     2
     0
    -1
```

```
if rank([A y])==rank(A)
disp('y is a column space of A')
elseif rank ([A; y'])==rank(A)
disp('y is a row space of A')
elseif A*y == 0
disp('y is in the null space of A')
elseif transpose(y)*A == 0
disp('y is in the left null space of A')
else
disp('y is not in any space of A')
end
```

```
y is in the null space of A
```

5e

```
A=[1 3 4 7;2 4 6 10;3 5 8 13;4 6 10 16]
```

```
A = 4×4
     1     3     4     7
     2     4     6    10
     3     5     8    13
     4     6    10    16
```

```
m=[-1;1;1;-1]
```

```
m = 4×1
    -1
     1
     1
    -1
```

```
if rank([A m])==rank(A)
disp('m is a column space of A')
elseif rank ([A; m'])==rank(A)
disp('m is a row space of A')
elseif A*m == 0
disp('m is in the null space of A')
elseif transpose(m)*A == 0
disp('m is in the left null space of A')
else
disp('m is not in any space of A')
end
```

```
m is in the left null space of A
```

6a

```
A=[1 -1 2 3;0 2 1 4;1 1 3 1;2 0 5 4]
```

```
A = 4×4
     1    -1     2     3
     0     2     1     4
     1     1     3     1
     2     0     5     4
```

```
v=[5;1;-2;0]
```

```
v = 4×1
     5
     1
    -2
     0
```

```
if rank([A v])==rank(A)
disp('v is a column space of A')
elseif rank ([A; v'])==rank(A)
disp('v is a row space of A')
elseif A*v == 0
disp('v is in the null space of A')
elseif transpose(v)*A == 0
disp('v is in the left null space of A')
else
disp('v is not in any space of A')
end
```

```
v is in the null space of A
```

6b

```
A=[1 -1 2 3;0 2 1 4;1 1 3 1;2 0 5 4]
```

```
A = 4×4
     1    -1     2     3
     0     2     1     4
     1     1     3     1
     2     0     5     4
```

```
w=[0;2;2;2]
```

```
w = 4×1
     0
     2
     2
     2
```

```
if rank([A w])==rank(A)
disp('w is a column space of A')
elseif rank ([A; w'])==rank(A)
disp('w is a row space of A')
elseif A*w == 0
disp('w is in the null space of A')
elseif transpose(w)*A == 0
disp('w is in the left null space of A')
```

```
else
disp('w is not in any space of A')
end
```

```
w is a column space of A
```

6c

```
A=[1 -1 2 3;0 2 1 4;1 1 3 1;2 0 5 4]
```

```
A = 4×4
     1    -1     2     3
     0     2     1     4
     1     1     3     1
     2     0     5     4
```

```
u=[-1;2;-1;1]
```

```
u = 4×1
    -1
     2
    -1
     1
```

```
if rank([A u])==rank(A)
disp('u is a column space of A')
elseif rank ([A; u'])==rank(A)
disp('u is a row space of A')
elseif A*u == 0
disp('u is in the null space of A')
elseif transpose(u)*A == 0
disp('u is in the left null space of A')
else
disp('u is not in any space of A')
end
```

```
u is not in any space of A
```

6d

```
A=[1 -1 2 3;0 2 1 4;1 1 3 1;2 0 5 4]
```

```
A = 4×4
     1    -1     2     3
     0     2     1     4
     1     1     3     1
     2     0     5     4
```

```
m=[3;-1;7;7]
```

```
m = 4×1
     3
    -1
     7
     7
```

```
if rank([A m])==rank(A)
disp('m is a column space of A')
elseif rank ([A; m'])==rank(A)
disp('m is a row space of A')
elseif A*m == 0
disp('m is in the null space of A')
elseif transpose(m)*A == 0
disp('m is in the left null space of A')
else
disp('m is not in any space of A')
end
```

```
m is a row space of A
```