LAB 2: Determinants and General Solution to Ax = b

(Math 250: Sections C3)

Name: Aryan Malhotra

Last 4 digits of RUID: 8256

Date: 11/04/2024

Learning Outcomes

- The determinant of a square matrix, how it changes under row operations and matrix multiplication.
- Geometry and Matrices
- The column space Col(A) of a matrix A.
- The null space Null(A) of a matrix A.
- Particular solutions and general solution to an inhomogeneous linear equation Ax = b.

Table of Contents

Random Seed

Initialize the random number generator by typing the following code, where abcd are the last four digits of your RUID

```
rng('default');
rng(abcd, 'twister');
```

```
% Enter your code here
rng('default');
rng(8256, 'twister');
```

This will ensure that you generate your own particular random vectors and matrices.

The lab report that you hand in must be your own work. The following problems use randomly generated matrices and vectors, so the matrices and vectors in your lab report will not be the same as those of other students doing the lab. Sharing of lab report files is not allowed in this course.

Question 1. The Determinant Function (Total Points - 4)

(a) **Cofactor Expansion:** The T-code ***cofactor.m*** calculates the matrix of cofactors of a square matrix. Generate a random 4×4 integer matrix ***a = rmat(4,4)***. Then use Matlab to calculate the cofactor matrix ***c = cofactor(a)***.

```
% Enter your code here
a = rmat(4,4)
```

```
a = 4×4
     6     9     3     0
     2     6     4     1
     1     1     3     8
     4     6     2     7
```

```
c = cofactor(a)
```

```
c = 4×4
   178  -100    70   -36
  -168   105    21     0
   126  -126   126     0
  -120   129  -147    54
```

Now use Matlab to calculate the four sums

```
a(1,1)*c(1,1) + a(1,2)*c(1,2) + a(1,3)*c(1,3) + a(1,4)*c(1,4)
a(2,1)*c(2,1) + a(2,2)*c(2,2) + a(2,3)*c(2,3) + a(2,4)*c(2,4)
a(1,3)*c(1,3) + a(2,3)*c(2,3) + a(3,3)*c(3,3) + a(4,3)*c(4,3)
a(1,4)*c(1,4) + a(2,4)*c(2,4) + a(3,4)*c(3,4) + a(4,4)*c(4,4)
```

```
% Enter your code here
a(1,1)*c(1,1) + a(1,2)*c(1,2) + a(1,3)*c(1,3) + a(1,4)*c(1,4)
```

```
ans =
378
```

```
a(2,1)*c(2,1) + a(2,2)*c(2,2) + a(2,3)*c(2,3) + a(2,4)*c(2,4)
```

```
ans =
378
```

```
a(1,3)*c(1,3) + a(2,3)*c(2,3) + a(3,3)*c(3,3) + a(4,3)*c(4,3)
```

```
ans =
378
```

```
a(1,4)*c(1,4) + a(2,4)*c(2,4) + a(3,4)*c(3,4) + a(4,4)*c(4,4)
```

```
ans =
378
```

(i) ***Exercise (Points - 1):*** Use cofactor expansion to explain why all sums give the same number. Check by using Matlab to calculate ***det(a)***.

Answer: For Cofactor Expansion we pick a certain row of choice in a matrix and loop over all entries in a row, adding the products of the entry and it's cofactor: $\sum_{j=1}^{n} a_{ij} \cdot c_{ij}$ ***for any row i in an*** $m \times n$ **matrix to get the Determinant of the Matrix (which is a fixed value).**

We do precisely that for each of the evaluations above hence each of them give us the same answer : The determinant of the Matrix.

```
% Enter your code here
det(a)
```

```
ans =
378
```

(b) Row Operations: Generate a 5×5 random integer matrix ***A = rmat(5,5)***.

```
% Enter your code here
A = rmat(5,5)
```

```
A = 5×5
     3     8     3     7     6
     3     4     7     0     5
     7     2     5     4     6
     5     3     5     7     4
     4     9     7     8     1
```

Use properties of the determinant function to answer the following:

Swap the first and second row of A to get the matrix B using the following commands:

```
B = A; B(2,:) = A(1,:); B(1,:) = A(2,:)
```

```
% Enter your code here
B = A; B(2,:) = A(1,:); B(1,:) = A(2,:)
```

```
B = 5×5
     3     4     7     0     5
     3     8     3     7     6
     7     2     5     4     6
     5     3     5     7     4
     4     9     7     8     1
```

(i) ***Exercise (Points - 1):*** What is the relation between det(A) and det(B)? Check your answer by calculating ***det(A)*** and ***det(B)*** using Matlab.

Answer: Row Swapping does not changes the sign of the determinant. Hence, det(A) = - det(B)

(this is because row switching corresponds to row addition followed by scalar multiplication by -1).

```
% Enter your code here
det(A)
```

```
ans =
-5.4870e+03
```

```
det(B)
```

```
ans =
5.4870e+03
```

Next, let C be the matrix obtained from A by multiplying the first row of A by 10 and adding to the second row of A using the following commands:

```
C = A; C(2,:) = A(2,:) + 10*A(1,:)
```

```
% Enter your code here
C = A; C(2,:) = A(2,:) + 10*A(1,:)
```

```
C = 5×5
     3     8     3     7     6
    33    84    37    70    65
     7     2     5     4     6
     5     3     5     7     4
     4     9     7     8     1
```

(ii) ***Exercise (Points - 1):*** What is the relation between det(A) and det(C)? Check your answer by Matlab.

Answer: since row addition does not change the determinant, and adding a scaled version of the row is just in some sense repeated row addition, the determinant does NOT change. Hence det(A) = det(C).

```
% Enter your code here
det(A)
```

```
ans =
-5.4870e+03
```

```
det(C)
```

```
ans =
-5.4870e+03
```

Finally, let D be the matrix obtained from A by multiplying the first row of A by 10:

```
D = A; D(1,:) = 10*A(1,:)
```

```
% Enter your code here
D = A; D(1,:) = 10*A(1,:)
```

```
D = 5×5
    30    80    30    70    60
     3     4     7     0     5
     7     2     5     4     6
     5     3     5     7     4
     4     9     7     8     1
```

(iii) ***Exercise (Points - 1):*** What is the relation between det(A), det(D), and det(10A)? Check your answers by Matlab.

Answer: Scalar multiplication of a row scales the determinant of the Matrix by the same factor

Hence, det(D) = 10 det(A)

```
% Enter your code here
det(D)
```

```
ans =
-54870
```

```
det(A)
```

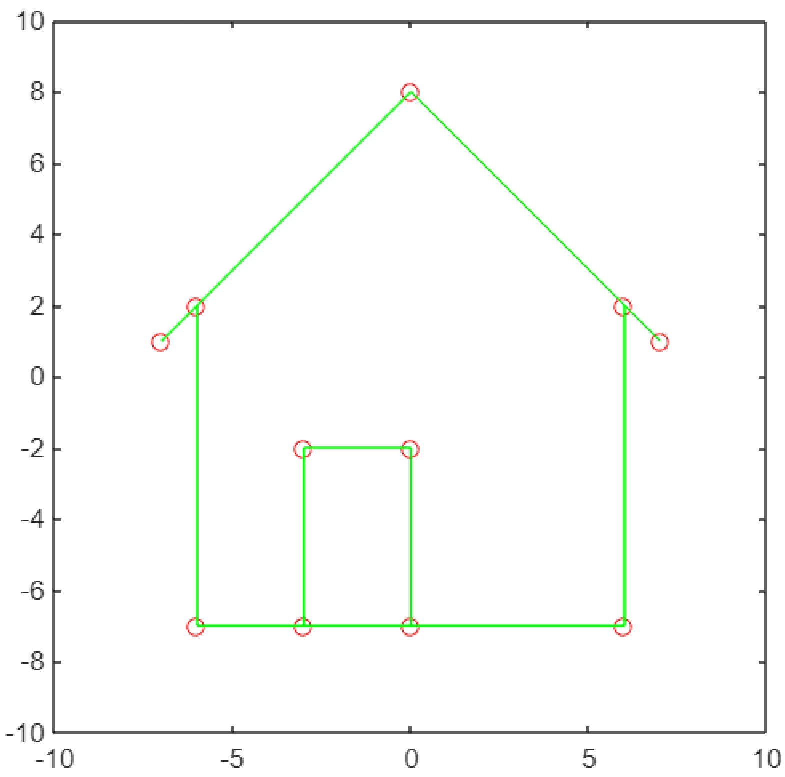
```
ans =
-5.4870e+03
```

Question 2. Geometry and Matrices (Total Points - 6)

This question uses Matlab to illustrate the geometric meaning of some special types of matrices. At the Matlab prompt type

```
H = house; plot2d(H), hold on
```

```
% Enter your code here
H = house; plot2d(H), hold on
```



A graphics window should open and display a crude drawing of a house. The matrix H contains the coordinates of the endpoints of the line segments making up the drawing

(a) **Rotations:** Generate a matrix Q by

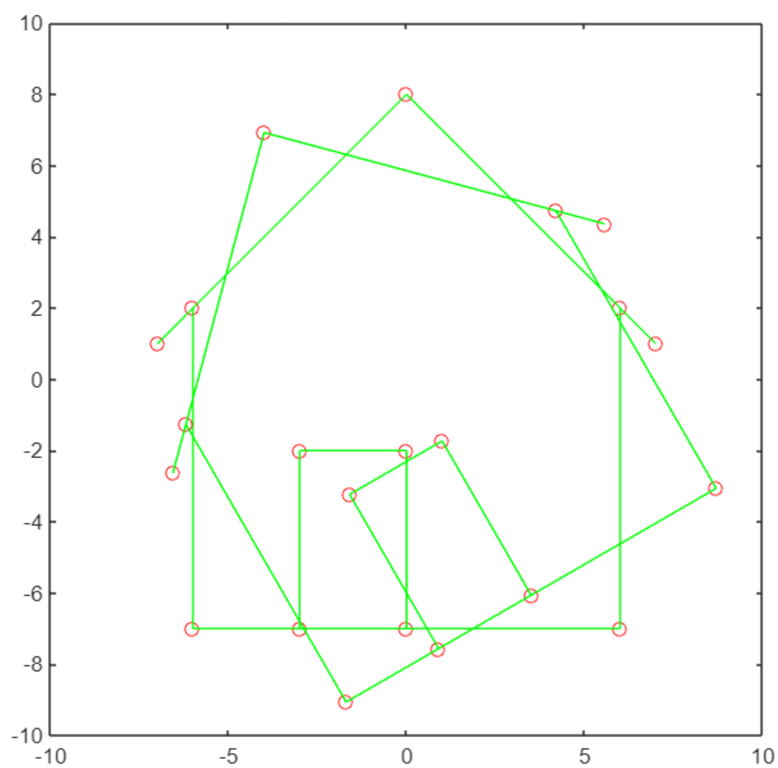
```
t = pi/6; Q = [cos(t), -sin(t); sin(t), cos(t)]
```

```
% Enter your code here
t = pi/6; Q = [cos(t), -sin(t); sin(t), cos(t)]
```

```
Q = 2×2
    0.8660   -0.5000
    0.5000    0.8660
```

Let Q act on the house by ***plot2d(Q*H)***.

```
% Enter your code here
plot2d(Q*H)
```



(i) ***Exercise (Points - 0.5):*** Describe in words how the house has changed.

Answer: The house rotated pi/6 radians anticlockwise about the origin

(ii) ***Exercise (Points - 0.5):*** Calculate ***det(Q)***. What does this tell you about the area inside the transformed house? (Use the properties and geometric interpretation of determinants.)

Answer: Since the det(Q) is 1. It transforms all areas in a space to areas of same magnitudes (by definition, determinants are how a transformation scales areas in a certain space). Hence, the area of the house would remain the same

```
% Enter your code here
det(Q)
```

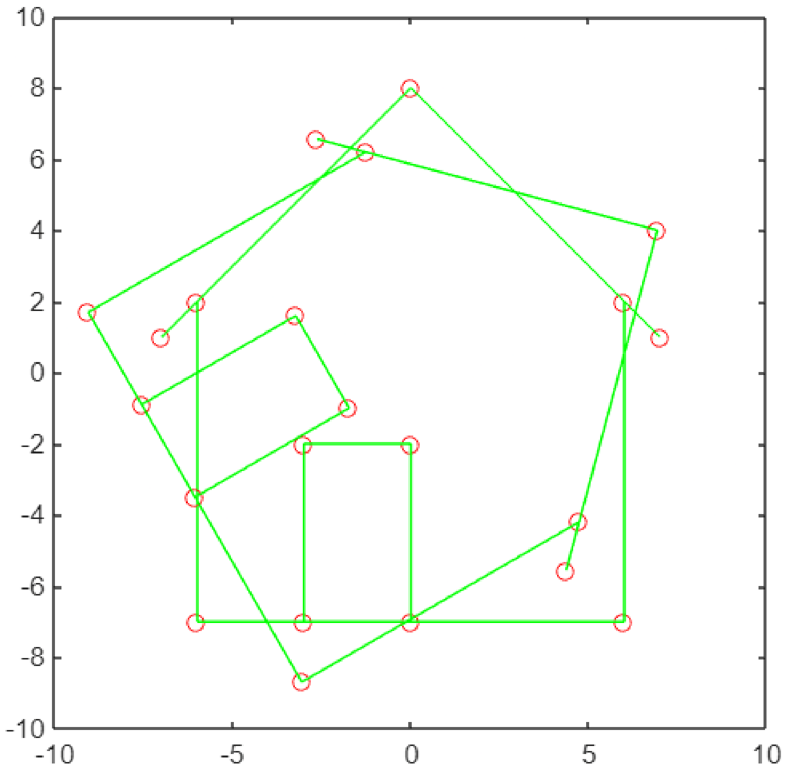
```
ans =
1
```

Clear the graphics window by the command ***clf*** and generate a new plot of the house as above.

```
% Enter your code here
figure(2); plot2d(H), hold on
```

Repeat the above process with t = - pi/3 and answer (i) and (ii) in this case.

```
% Enter your code here
t = -pi/3; Q = [cos(t), -sin(t); sin(t), cos(t)]; plot2d(Q*H)
```



(i) ***Exercise (Points - 0.5):*** Describe in words how the house has changed.

Answer: This time, the house rotated pi/3 radians clockwise about the origin.

(ii) ***Exercise (Points - 0.5):*** Calculate ***det(Q)***. What does this tell you about the area inside the transformed house?

Answer: the det(Q) is still 1. Again, that means all areas are scaled by the factor of 1 upon the transformation.

```
% Enter your code here
det(Q)
```

```
ans =
1
```

(b) Dilations: Clear the graphics window and generate a new plot of the house as above.

```
% Enter your code here
figure(3); plot2d(H), hold on
```

Generate a matrix D by

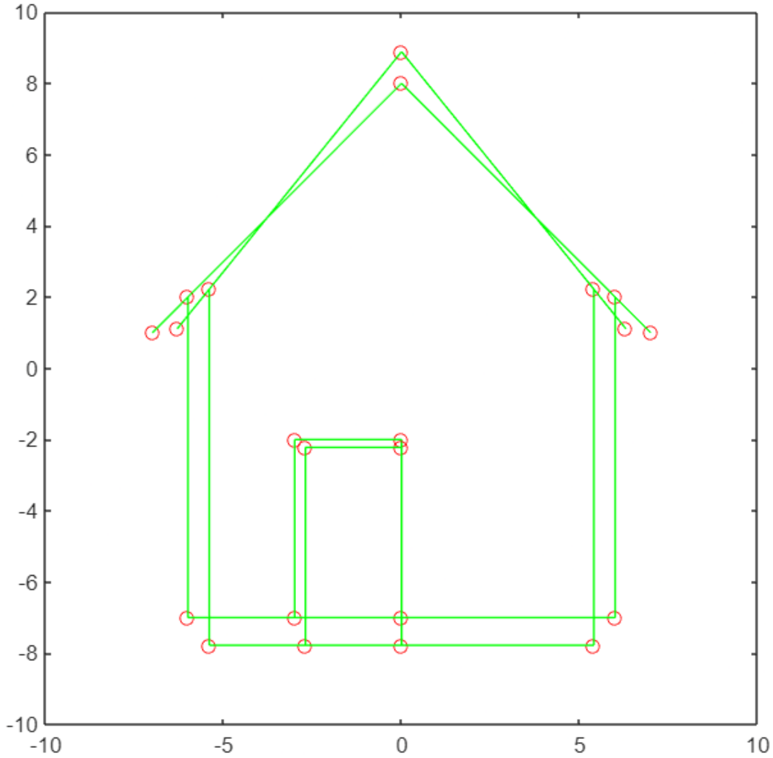
```
r = .9; D = [r, 0; 0, 1/r]
```

```
% Enter your code here
r = .9; D = [r, 0; 0, 1/r]
```

```
D = 2×2
    0.9000         0
         0    1.1111
```

Let D act on the house by ***plot2d(D*H)***.

```
% Enter your code here
plot2d(D*H)
```



(i) ***Exercise (Points - 0.5):*** Describe in words how the house has changed.

Answer: The house has been stretched vertically. All points in the x direction are now closer to the origin and all of them in the y direction have been scaled away from the origin.

(ii) ***Exercise (Points - 0.5):*** Calculate ***det(D***). What does this tell you about the area inside the transformed house?

Answer: Since det(D) is still 1, that means the area of the house is still the same since by definition, the determinant is the amount that a linear transformation would scale any area by.

```
% Enter your code here
det(D)
```

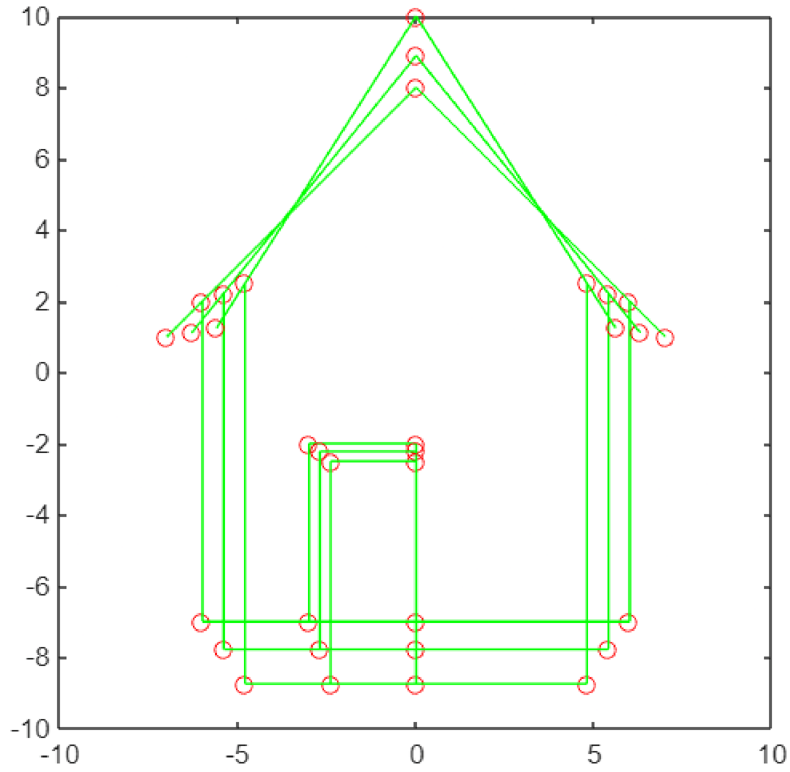
```
ans = 
1
```

Repeat the above process with r = .8 and answer (i) and (ii) in this case.

```
% Enter your code here
r = .8; D = [r, 0; 0, 1/r]
```

```
D = 2×2
    0.8000         0
         0    1.2500
```

```
plot2d(D*H)
```



(i) ***Exercise (Points - 0.5):*** Describe in words how the house has changed.

Answer: The house has now been stretched even more in the verticle direction. It is the same as r=0.9 in essense but the intensity of the stretch has been increased.

(ii) ***Exercise (Points - 0.5):*** Calculate ***det(D)***. What does this tell you about the area inside the transformed house?

Answer: since det(D), yet again is 1, the area of the transformed shape would still be the same as that of the house initially.

```
% Enter your code here
det(D)
```

```
ans =
1
```

(c) Shearing Transformations: Clear the graphics window and generate a new plot of the house as above.

```
% Enter your code here
figure(4); plot2d(H), hold on
```

Generate a matrix T by

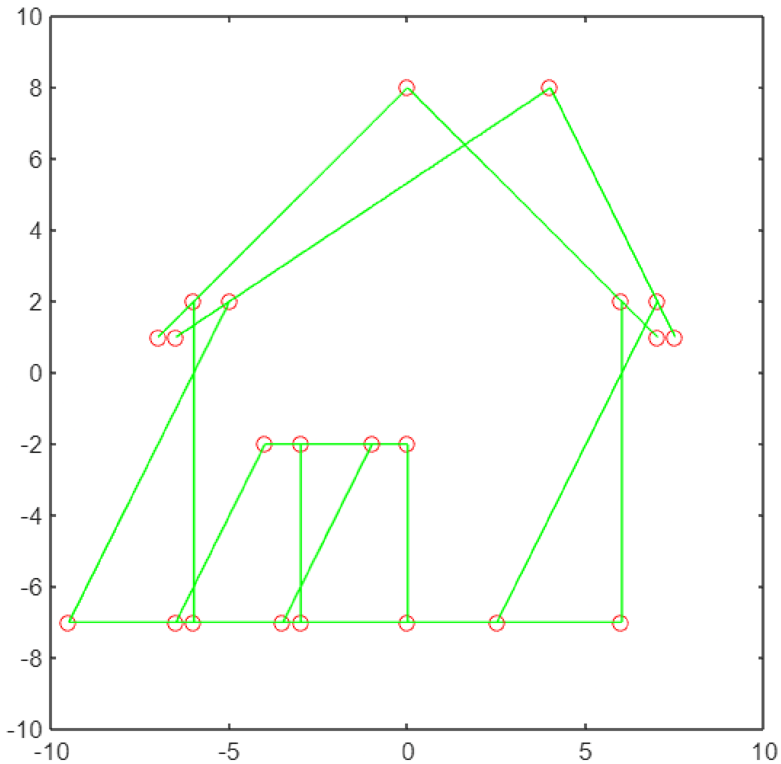
```
t = 1/2; T = [1, t; 0, 1]
```

```
% Enter your code here
t = 1/2; T = [1, t; 0, 1]
```

```
T = 2×2
    1.0000    0.5000
         0    1.0000
```

Let T act on the house by ***plot2d(T*H)***.

```
% Enter your code here
plot2d(T*H)
```



(i) ***Exercise (Points - 0.5):*** Describe in words how the house has changed.

Answer: The house seems to have been skewed horizontally. The verticle position of the points seems to have not changed but everything seems to have been streched in the horontal direction.

(ii) ***Exercise (Points - 0.5):*** Calculate ***det(T)***. What does this tell you about the area inside the transformed house?

Answer: det(T)=1 implies that the area of the house, yet again, remains the same.

```
% Enter your code here
det(T)
```

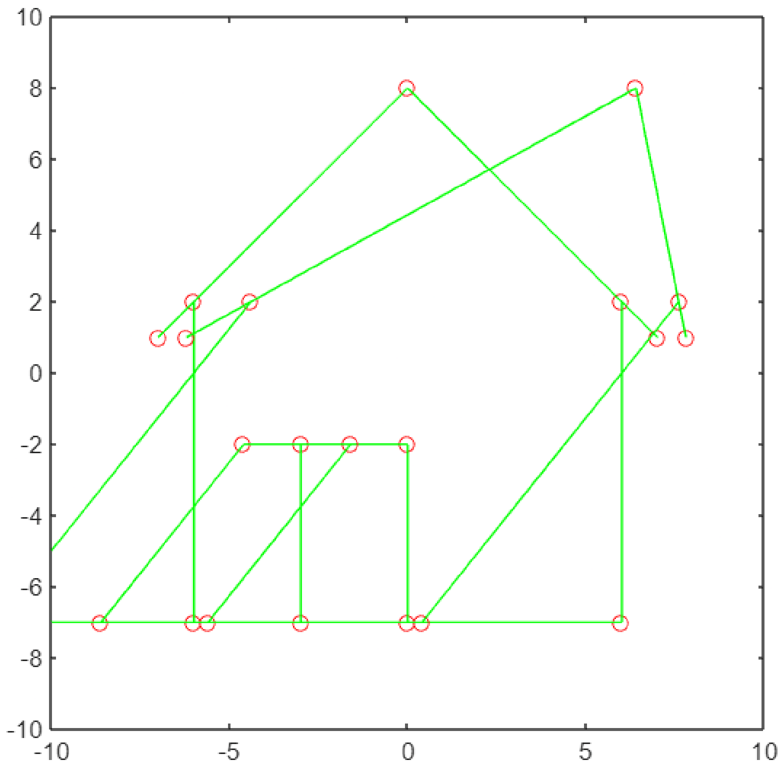
```
ans =
1
```

Repeat the above process with r = .8 and answer (i) and (ii) in this case.

```
% Enter your code here
figure(5); plot2d(H), hold on
t = .8; T = [1, t; 0, 1]
```

```
T = 2×2
    1.0000    0.8000
         0    1.0000
```

```
plot2d(T*H)
```



(i) ***Exercise (Points - 0.5):*** Describe in words how the house has changed.

Answer: The effect of the skewness has now been increases from what was there in the previous example. The verticle components of each vector still remain the same, however, the horizontal components have been drastically stretched.

(ii) ***Exercise (Points - 0.5):*** Calculate ***det(T)***. What does this tell you about the area inside the transformed house?

Answer: Again, det(T)=1 implying that the area of the house remains the same.

```
% Enter your code here
```

```
det(T)
```

```
ans =
1
```

Question 3. Visualizing the Column Space (Total Points - 4)

In this question you will determine visually whether given vectors lie in the column space of a matrix.

Now generate a random 3×2 integer matrix by the Matlab command ***A = rmat(3,2)*** and calculate ***rank(A)***. Since A is a random matrix, the rank is very likely to be 2. If the rank is not 2, generate another A. Repeat the test until you get a matrix with rank 2. Use this matrix in the rest of the question. *If you need to generate more than one matrix, include all the matrices you generate in your lab report.*

```
% Enter your code here
A = rmat(3,2)
```

```
A = 3×2
     8     9
     9     6
     1     0
```

```
rank(A)
```

```
ans =
2
```

(a) ***Exercise (Points - 1):*** Define ***u = A(:,1), v = A(:,2)*** to be the column vectors for A.

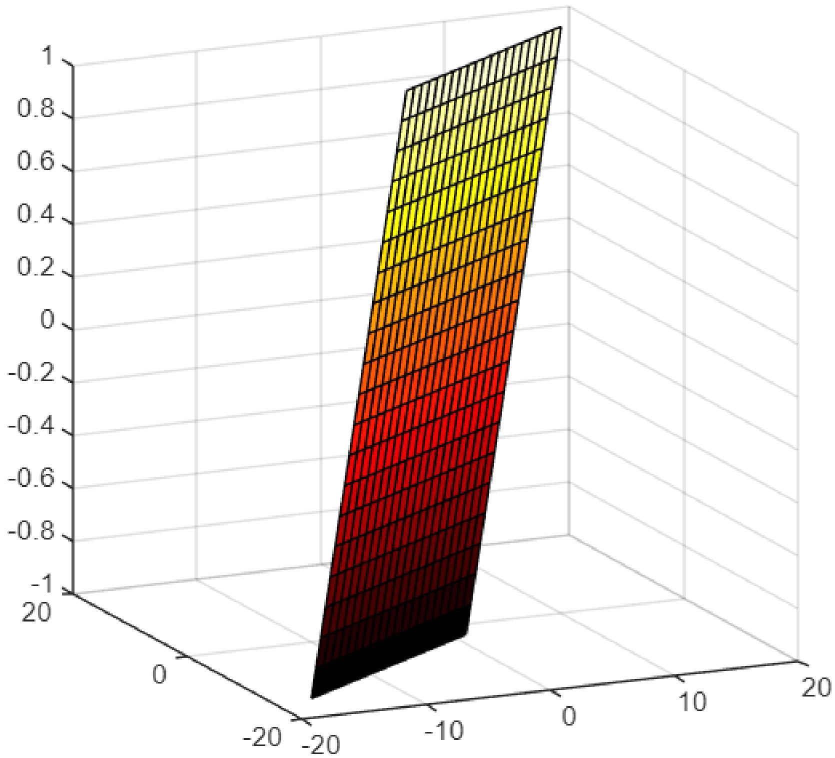
```
% Enter your code here
u = A(:,1), v = A(:,2)
```

```
u = 3×1
     8
     9
     1
v = 3×1
     9
     6
     0
```

To clear the graphics window and graph the column space Col(A) of A, enter the Matlab commands

```
clf
[s,t] = meshgrid((-1:0.1:1), (-1:0.1:1));
X = s*u(1)+t*v(1); Y = s*u(2)+t*v(2); Z = s*u(3)+t*v(3);
surf(X,Y,Z); axis square; colormap hot, hold on
```

```
% Enter your code here
clf
[s,t] = meshgrid((-1:0.1:1), (-1:0.1:1));
X = s*u(1)+t*v(1); Y = s*u(2)+t*v(2); Z = s*u(3)+t*v(3);
surf(X,Y,Z); axis square; colormap hot, hold on
```



A graph should appear in a separate window showing Col(A). From the ***Tools*** menu choose the command ***Rotate 3D***. Using the mouse, position the cursor over the graph. Press and hold the left mouse button until a box appears to enclose the graph. Then move the mouse to rotate the graph in three dimensions.

(b) Generate a random vector in $\mathbb{R}^3$ using ***b = rvect(3).***

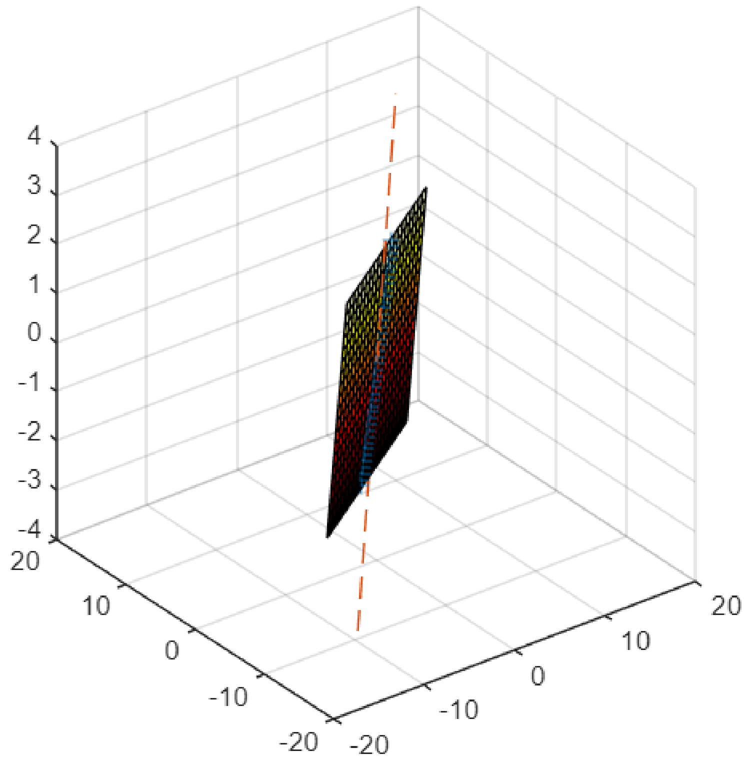
```
% Enter your code here
b = rvect(3)
```

```
b = 3×1
     9
     9
     4
```

To graph the line Span(b) in the same figure as Col(A), enter the commands

```
r = -1:0.05:1; plot3(r*b(1),r*b(2),r*b(3), '--')
```

```
% Enter your code here
r = -1:0.05:1; plot3(r*b(1),r*b(2),r*b(3), '--')
```



(i) ***Exercise (Points - 0.5):*** Now determine whether b lies inside Col(A) graphically. By manually rotating (using your mouse) the plot enough, you should be able to see whether the entire line Span(b) lies in Col(A) or not.

Answer: No, the point b and it's Span - Span(b) do not entirely lie within the Col(A)

Important: For every vector v, the line Span(v) will intersect Col(A) in the point 0, since every subspace contains 0. You must look to see if all of the line through your vector b is in Col(A). (Hint: Try to make Col(A) look like a line by viewing it edge-on.)

(ii) ***Exercise (Points - 0.5):*** Save the graph with a good choice of rotation showing whether or not b is in Col(A).

(iii) ***Exercise (Points - 1):*** Can you find a vector $x \in \mathbb{R}^2$ such that Ax = b, where A is the matrix and b is the vector that you have generated? Explain why or why not using the graph from part (b).

Answer: The Matrix A transforms vectors x to vectors in Col(A) since entries in x act as weights for columns of A. Hence, each resultant vector b must be a linear combination of columns of A i.e. it must be in Col(A).

But since our choice of b is NOT in Col(A), there is no solution vector **$x \in \mathbb{R}^2$ such that Ax = b.**

Generate a random vector lying in Col(A) using the commands

```
z = rand(2,1), c = A*z
```

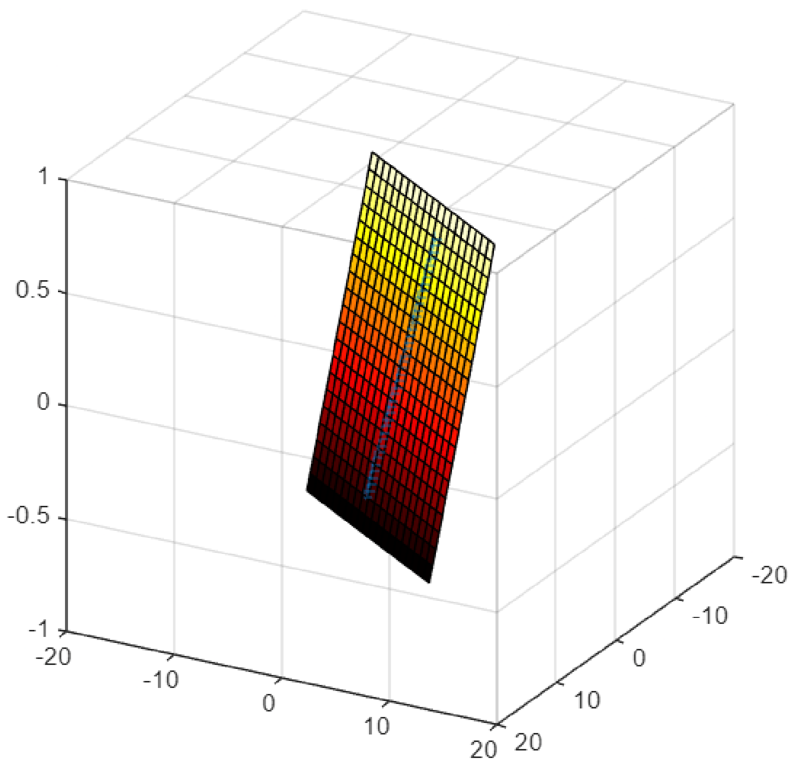
```
% Enter your code here
z = rand(2,1), c = A*z
```

```
z = 2×1
    0.8003
    0.1419
c = 3×1
    7.6792
    8.0538
    0.8003
```

Plot a new graph of Span(c) and Col(A) using

```
figure, surf(X,Y,Z); axis square; colormap hot, hold on plot3(r*c(1),r*c(2),r*c(3), '+')
```

```
% Enter your code here
figure, surf(X,Y,Z); axis square; colormap hot, hold on,
plot3(r*c(1),r*c(2),r*c(3), '+')
```



(vi) ***Exercise (Points - 1):*** Manually rotating (using your mouse) the plot to show that the entire line Span(c) is contained in Col(A) and save the graph with a good choice of rotation.

Question 4. Reduced Row Echelon Form and Null Space (Total Points - 5)

Generate a partly random 3×5 matrix A and its reduced row echelon form R. First generate a random 3×3 integer matrix and check its rank:

```
B = rmat(3,3), rank(B)
```

```
% Enter your code here
B = rmat(3,3), rank(B)
```

```
B = 3×3
     7     6     0
     7     1     2
     3     7     0
ans =
3
```

Since B is random, it is very likely to have rank 3. If not, generate another B until this is true. Keep all the matrices you generate in your lab report.

Now use B to define a 3×5 matrix A and its reduced row echelon form R by

```
A = [B(:,1), B(:,2), 2*B(:,1) + 3*B(:,2), 4*B(:,1) - 5*B(:,2), B(:,3)], R = rref(A)
```

```
% Enter your code here
A = [B(:,1), B(:,2), 2*B(:,1) + 3*B(:,2), 4*B(:,1) - 5*B(:,2), B(:,3)], R =
rref(A)
```

```
A = 3×5
     7     6    32    -2     0
     7     1    17    23     2
     3     7    27   -23     0
R = 3×5
     1     0     2     4     0
     0     1     3    -5     0
     0     0     0     0     1
```

(a)

(i) ***Exercise (Points - 1):*** Explain why columns #1, #2, and #5 are the pivot columns of A and R. (Hint: Use the theorem (nonpivot columns of a matrix are linear combinations of its pivot columns).)

Answer: We defined columns #1, #2, #5 of matrix A to be columns of matrix B which were linearly independent. Columns #3 and #4, by definition were 3*B_2 and 4*B_1 (B_i being ith column of B). Hence we defined them to be linear combinations of other (pivot) columns of matrix A. Hence, they are non pivot columns with independent variables.

(ii) ***Exercise (Points - 1):*** Let V be the set of solutions to the homogeneous system of equations Ax = 0 (the null space of A). In the equation Ax = 0 (where x ∈ $\mathbb{R}^5$), what are the free variables and what is dim(V)?

Answer: Since Col 3 and Col 4 are non pivot columns, x3 and x4 are going to be the free variables for solution x = [x1,x2,x3,x4,x5].

dim(V) is gonna be # Non Pivot Columns = 2

(b) Use the Matlab T-codes ***nulbasis.m*** to calculate the special solutions to the system of equations Ax = 0:

```
N = nulbasis(A)
```

```
% Enter your code here
N = nulbasis(A)
```

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

The columns of N are the solutions to Ax = 0 obtained by setting one free variable to 1 and all the other free variables to 0. (the vectors that multiply the free variables in the solution of Ax=0 in the parametric vector form). Define

```
v1 = N(:,1), v2 = N(:,2)
```

```
% Enter your code here
v1 = N(:,1), v2 = N(:,2)
```

```
v1 = 5×1
    -2
    -3
     1
     0
     0
v2 = 5×1
    -4
     5
     0
     1
     0
```

(Notice that v1 and v2 are vectors with 5 components, not scalars.)

(i) ***Exercise (Points - 0.5):*** Which component of v1 is 1 and which components of v1 are zero?

Answer: the 3rd entry/component of v2 is 1 and the 3th and 5th enties/components of v2 are zero

(ii) ***Exercise (Points - 0.5):*** Which component of v2 is 1 and which components of v2 are zero?

Answer: the 4th entry/component of v1 is 1 and the 4th and 5th enties/components of v1 are zero

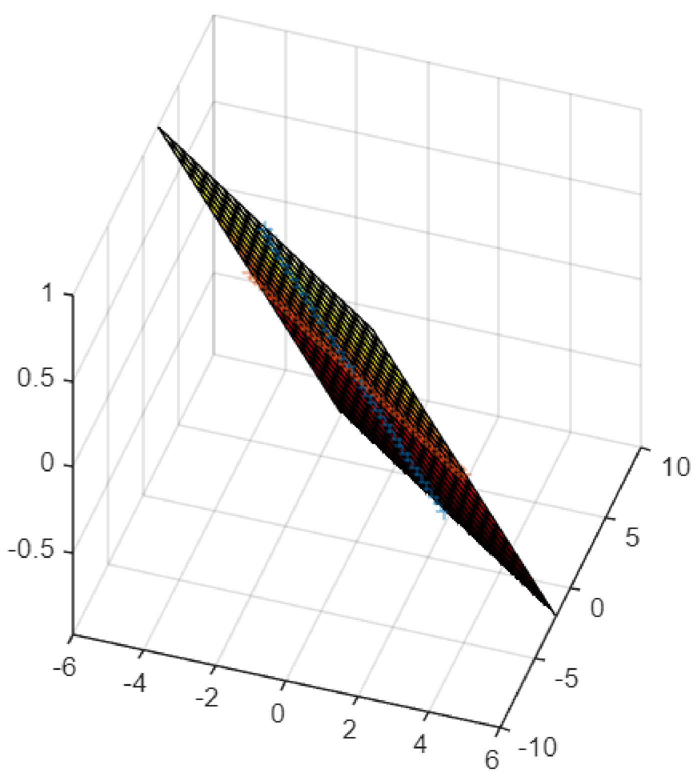
(iii) ***Exercise (Points - 1):*** Check by Matlab that v1 and v2 are in Null(A).

since the rank of the matrix doesn't change if we add vectors v1 and v2, the vectors do not add any additional points in space and are hence within the vector space

```
% Enter your code here
rank(N), rank([N(:,1), N(:,2), v1]), rank([N(:,1), N(:,2), v2])
```

```
ans =
2
ans =
2
ans =
2
```

```
clf
[s,t] = meshgrid((-1:0.1:1), (-1:0.1:1));
X = s*N(1,1)+t*N(1,2); Y = s*N(2,1)+t*N(2,2); Z = s*N(3,1)+t*N(3,2);
figure, surf(X,Y,Z); axis square; colormap hot, hold on,
plot3(r*v1(1),r*v1(2),r*v1(3), '+'), plot3(r*v2(1),r*v2(2),r*v2(3), '+')
```



(c) Now generate a random linear combination x of the vectors v1 and v2 by

```
s = rand(1), t = rand(1), x = s*v1 + t*v2
```

```
% Enter your code here
s = rand(1), t = rand(1), x = s*v1 + t*v2
```

```
s =
```

```
0.0971
t =
0.8235
x = 5×1
   -3.4881
    3.8259
    0.0971
    0.8235
         0
```

(Note that each occurrence of ***rand(1)*** generates a different random coefficient).

(i) ***Exercise (Points - 1):*** Explain (without Matlab) why x satisfies Ax = 0. Also explain why x satisfies Rx = 0. Then confirm this by Matlab.

Answer: Since v1 and v2 are the basis vectors for the Null(A), the space they span is the Null space of A, i.e. space of all vectors that satisfies Ax=0. Any linear combination of v1 and v2 is within this solution set/space. Since x is a linear combination of our Null Space basis, it is within the Null space as well, hence satisfies Ax=0

Question 5. Particular Solution and General Solution to Ax = b (Total Points - 6)

Let A be a matrix of size $m \times n$ with $m \neq n$. The linear system Ax = b is called *underdetermined* if $m < n$ (more variables than equations). In this case there are more columns than pivots, so there are always free variables. Hence a solution x is never unique (the solution might or might not exist, depending on the choice of b). The system is called *overdetermined* if $m > n$ (more equations than variables). In this case, there are more rows than pivots, and hence there are always vectors b$\neq$0 for which there is no solution x. In both cases the matrix A is not square, so the system can never be solved by finding an inverse matrix for A.

(a) **Particular Solution (overdetermined system):** Generate a random 5×3 integer matrix A (the coefficient matrix for an *overdetermined* system of 5 equations in 3 unknowns) and its reduced row echelon form R by

```
A = rmat(5, 3), R = rref(A)
```

```
% Enter your code here
A = rmat(5, 3), R = rref(A)
```

```
A = 5×3
     8     4     4
     6     3     4
     3     7     6
     9     7     7
     0     1     7
R = 5×3
     1     0     0
     0     1     0
     0     0     1
     0     0     0
     0     0     0
```

Since A is random, the matrix A is very likely to have rank 3. If the rank of A is not 3, generate a new matrix A until the rank of A is 3, and use this matrix. Keep all the matrices you generate in your lab report.

(i) ***Exercise (Points - 1):*** Explain (without Matlab) why there exist vectors b $\in \mathbb{R}^5$ such that equation Ax = b does not have a solution. (Hint: the test for consistency).

Answer: Solving for Ax=b is like finding a row equivalent form of [A|b] <-> [RREF(A)|c]

But RREF(A) has 2 [0,0,0] rows. Hence, [RREF(A)|C] may have [0 0 0| d] rows, d being non zero. Hence, the system would be inconsistent since 0+0+0 can never be nonzero.

(Another way that I conceptualize this is that when #rows > #columns, since each column corresponds to a degree of freedom in input space and each row to that in output space, our output space > input space. Hence, there must be some part of the output space that remains unmapped by a linear tranformation of the input space.)

The following Matlab command will generate a random 5×1 vector b and try to find a particular solution to Ax = b:

```
b = rmat(5,1), xp = partic(A, b)
```

```
% Enter your code here
b = rmat(5,1), xp = partic(A, b)
```

```
b = 5×1
     2
     6
     6
     1
     1
xp =

     []
```

The answer should be *xp = []* (empty vector, meaning no solution).

Now use the (partly) random vector

```
b = rand(1)*A(:,1) + rand(1)*A(:,2) + rand(1)*A(:,3)
```

and calculate ***xp = partic(A, b)***. Then use Matlab to check that A*xp = b.

```
% Enter your code here
b = rand(1)*A(:,1) + rand(1)*A(:,2) + rand(1)*A(:,3), xp = partic(A, b)
```

```
b = 5×1
    9.1874
    7.2310
   10.2556
   13.5862
```

```
    3.3424
xp = 3×1
    0.4984
    0.9597
    0.3404
```

(ii) ***Exercise (Points - 1):*** Explain (without Matlab) why the special form of this b guarantees that there is a solution to Ax = b.

Answer: b vector, as we define it is a linear combination of the columns of Matrix A. That means matix A's columns can be weighed by certain values to get vector be. That is precisely what Ax=b represents. x here is a vector of all the weights of colums of A so as to generate b. Since that is how we generate b in the first place, the solution must exist.

(b) **Particular Solution (underdetermined system):** Generate a random 3×5 integer matrix A (the coefficient matrix for an *underdetermined system* of 3 equations in 5 unknowns) and its reduced row echelon form R:

```
% Enter your code here
A = rmat(3,5), R = rref(A)
```

```
A = 3×5
     7     2     8     0     9
     4     9     5     4     0
     0     1     9     1     7
R = 3×5
    1.0000         0         0   -0.2073    0.5081
         0    1.0000         0    0.5061   -0.7012
         0         0    1.0000    0.0549    0.8557
```

Since A is random, the matrix A(:, 1 : 3) is very likely to have rank 3. If the rank of A(:, 1 : 3) is not 3, generate a new matrix A until the rank of A(:, 1 : 3) is 3, and use this matrix. As usual, keep all the matrices you generate in your lab report.

(i) ***Exercise (Points - 1):*** Explain (without Matlab) why the equation Ax = b has a solution for every vector b $\in \mathbb{R}^3$. (Hint: the test for consistency).

Answer: there is NO row for [A|b] such that we get a [0 0 0 | c] row. Hence, the matrix equation Ax=b will always have a solution.

Now generate a random 3×1 vector b and use the T-code ***partic.m*** to find a particular solution to Ax = b by

```
b = rmat(3,1), xp = partic(A, b)
```

```
% Enter your code here
b = rmat(3,1), xp = partic(A, b)
```

```
b = 3×1
     8
     8
     0
```

```
xp = 5×1
    1.0732
    0.4390
   -0.0488
         0
         0
```

This is the solution with all the free variables set to zero (the constant vector term in the solution of Ax=b in the parametric vector form).

(ii) ***Exercise (Points - 1):*** Why are the entries in row 4 and 5 of **xp** zero? Check by Matlab that A*xp = b.

Answer: Since the 4th and 5th enties correspond to free variable terms in the parametric vector form, the 4th and 5th entires could have taken any value. The software sets the free variables to zero hence the entries are zero. Also, since the solution is the constant term, the vector in free space/null space has to be the zero vector.

```
% Enter your code here
A*xp,b
```

```
ans = 3×1
     8
     8
     0
b = 3×1
     8
     8
     0
```

(c) **General Solution (underdetermined system):** Let A be the 3×5 matrix and b the 3×1 vector that you generated in part (b). The general solution to an inhomogeneous linear system Ax = b is obtained by adding a vector from the *null space* of A to the particular solution **xp**.

(i) ***Exercise (Points - 1):*** Use the T-code ***nulbasis.m*** to obtain the 5×2 matrix

```
N = nulbasis(A)
```

```
% Enter your code here
N = nulbasis(A)
```

```
N = 5×2
    0.2073   -0.5081
   -0.5061    0.7012
   -0.0549   -0.8557
    1.0000         0
         0    1.0000
```

Set ***v1 = N(:,1), v2 = N(:,2)*** and form a random *general solution* to Ax = b

```
x = xp + rand(1)*v1 + rand(1)*v2
```

```
% Enter your code here
v1 = N(:,1), v2 = N(:,2), x = xp + rand(1)*v1 + rand(1)*v2
```

```
v1 = 5×1
    0.2073
   -0.5061
   -0.0549
    1.0000
         0
v2 = 5×1
   -0.5081
    0.7012
   -0.8557
         0
    1.0000
x = 5×1
    1.1696
    0.1057
   -0.2544
    0.9106
    0.1818
```

Check by Matlab that A*x is the vector b.

```
% Enter your code here
A*x, b
```

```
ans = 3×1
     8
     8
     0
b = 3×1
     8
     8
     0
```

(ii) ***Exercise (Points - 1):*** Now solve the equation Ax = b with the extra condition that x should be of the form $x = [\ x_1,\ x_2,\ x_3,\ -9,\ 8\ ]^T$.

For this, you must choose particular scalars c1 and c2 in the general solution ***x = xp + c1*v1 + c2*v2***. (Hint: Look at the free variables.) Then check by Matlab that A*x is the vector b.

Answer: the 4th and 5th entries of xp are 0. v1 corresponds to the vector multiplied by the free variable x4 and v2 corresponds to the vector multiplied by the free variable x5. i.e. x = [x1 x2 x3 x4 x5] = [x1' x2' x3' 0 0] + x4v1 + x5v2 but x = xp + c1*v1 + c2*v2

Hence, c1 = x4 and x2=x5

hence, c1 = -9 and c2 = 8 (another way to look at this is, this is the only way 4th element is -9 and 8th element is 8)

```
% Enter your code here
c1=-9; c2=8; x = xp + c1*v1 + c2*v2;
A*x, b
```

```
ans = 3×1
```

```
      8
      8
      0
b = 3×1
      8
      8
      0
```

Helper Function (Do Not Edit)

```
function v = rvect(m)
v = fix(10*rand(m,1));
end
```

```
function A = rmat(m,n)
A = fix(10*rand(m,n));
end
```