Table of Contents


```
%%%% MATLAB PROJECT 3, MATH461, LINEAR ALGEBRA FOR SCIENTIST AND ENGINEERS
%%%% Romeo Perlstein, 4/10/2023
%%%% UID: 118030685, section 0123
```

Question 1

QUESTION 40 FROM 4.4 IN "LINEAR ALGEBRA AND IT'S APPLICATIONS - 6tH EDITION:

Let H = Span {v1, v2 , v3} and B = {v1, v2 , v3 }. Show that B is a basis for H and x is in H , and find the B-coordinate vector of x, for (The following vectors)

```
v1_1 = [-6 ; 4 ; -9 ; 4]
v2_1 = [8 ; -3 ; 7; -3]
v3_1 = [-9;5;-8;3]
x_1 = [4;7;-8;3]

% Augment v1 v2 v3 and REF
H = [v1_1 v2_1 v3_1]
R_1 = rref(H)
if (R_1(1,1) == 1)
    if (R_1(2,2) == 1)
        if (R_1(3,3) == 1)
            fprintf("Matrix has 3 pivots!\n\n")
```

```
            fprintf("B is a basis for H\n")
            fprintf("since basis colH = {v1, v2, v3}\n")

        end
    end
end
% x is in H if there is a linear combination such that Ha = x
% augment matrix again
x_in_H = [H x_1]
weights_for_x = rref(x_in_H)

if (weights_for_x(1,1) == 1)
    fprintf("x1 = ")
    fprintf(string(weights_for_x(1,4)))
    fprintf("\n")

    if (R_1(2,2) == 1)
        fprintf("x2 = ")
        fprintf(string(weights_for_x(2,4)))
        fprintf("\n")

        if (R_1(3,3) == 1)
        fprintf("x3 = ")
        fprintf(string(weights_for_x(3,4)))
        fprintf("\n")
        fprintf("x4=0, so every row has a pivot\n")
        fprintf("AKA, there is a linear combination")
        fprintf("such that Ha = x\n")
        end
    end
end
% Every Row has a pivot when row reduction the Augmented Matrix, meaning
% that theres a linear combination of H and weights such that Ha = x,
% meaning that x is H

% [x]B = c1[-6;4;-9;4] + c2[8;-3;7;-3] + c3[-9;5;-8;3] = [4;7;-8;3]
% make an augmented matrix using c1, c2, c3, which is the same as the
% augmented matrix from the first half of the problem
c1 = weights_for_x(1,4)
c2 = weights_for_x(2,4)
c3 = weights_for_x(3,4)
x_coords_in_B = [c1;c2;c3]

x_coords_check = c1*v1_1 + c2*v2_1 + c3*v3_1

if (x_coords_check == x_1)
    fprintf("Check is all good, c1, c2, c3 are the\n")
    fprintf("B coords of x\n")
end
```

```
v1_1 =

     -6
```

```
      4
     -9
      4


v2_1 =

      8
     -3
      7
     -3


v3_1 =

     -9
      5
     -8
      3


x_1 =

      4
      7
     -8
      3


H =

     -6              8             -9
      4             -3              5
     -9              7             -8
      4             -3              3


R_1 =

      1              0              0
      0              1              0
      0              0              1
      0              0              0

Matrix has 3 pivots!

B is a basis for H
since basis colH = {v1, v2, v3}

x_in_H =

     -6              8             -9              4
      4             -3              5              7
     -9              7             -8             -8
```

```
        4              -3               3               3


weights_for_x =

        1               0               0               3
        0               1               0               5
        0               0               1               2
        0               0               0               0

x1 = 3
x2 = 5
x3 = 2
x4=0, so every row has a pivot
AKA, there is a linear combinationsuch that Ha = x

c1 =

        3


c2 =

        5


c3 =

        2


x_coords_in_B =

        3
        5
        2


x_coords_check =

        4
        7
       -8
        3

Check is all good, c1, c2, c3 are the
B coords of x
```

Question 2

```
format short

% set_of_func = {1, cos(t), cos(t)^2, cos(t)^3}
```

a)

```
t1_2 = 0
t2_2 = .1
t3_2 = .2
t4_2 = .3
A_2 = [1 cos(t1_2) cos(t1_2)^2 cos(t1_2)^3 ; 1 cos(t2_2) cos(t2_2)^2
 cos(t2_2)^3; 1 cos(t3_2) cos(t3_2)^2 cos(t3_2)^3; 1 cos(t4_2) cos(t4_2)^2
 cos(t4_2)^3]
```

b)

```
rref_A_2 = rref(A_2)
detA_2 = det(A_2)
```

c)

The last two computations show that A is invertible because according to the invertible matrix theorem, in order for a matrix to be invertible it must have a pivot in every row and have a non-zero determinent. The solutions above show that A has a pivot in every row, and also that it's Determinant is non-zero

d)

```
t1_2 = 0
t2_2 = .2
t3_2 = .5
t4_2 = 1
A_2_check = [1 cos(t1_2) cos(t1_2)^2 cos(t1_2)^3 ; 1 cos(t2_2) cos(t2_2)^2
 cos(t2_2)^3; 1 cos(t3_2) cos(t3_2)^2 cos(t3_2)^3; 1 cos(t4_2) cos(t4_2)^2
 cos(t4_2)^3]
rref(A_2_check)
det(A_2_check)

% While definitely larger, still quite small to eliminate total suspicion.
% I would've never been suspicious though if the problem didn't outright
% mention it
```

e)

(I like this question) The set of functions {1, sin^2(t), and cos^2(t) are linearly dependent because of the trig identity that sin^2(t) + cos^2(t) = 1, and is able to be rearranged to match the problem statement that means that there is a non-trivial set of solutions in which x1(1) + x2(sin^2(t)) + x3(cos^2(t)) = 0, where x1, x2, x3 are non-zero. In fact, the solution set is x1 = -1, x2 = 1, and x3 = 1 => -1 + sin^2 + cos^2 = 0

f)

```
t1_2 = 0
t2_2 = .1
t3_2 = .2
A_2_f = [1 sin(t1_2)^2 cos(t1_2)^2 ; 1 sin(t2_2)^2 cos(t2_2)^2 ; 1 sin(t3_2)^2
 cos(t3_2)^2]
```

```
rref(A_2_f)
det(A_2_f)

% checking again
t1_2 = 0
t2_2 = 2
t3_2 = 4
A_2_f = [1 sin(t1_2)^2 cos(t1_2)^2 ; 1 sin(t2_2)^2 cos(t2_2)^2 ; 1 sin(t3_2)^2
 cos(t3_2)^2]
rref(A_2_f)
det(A_2_f)

% Det is still basically 0
```

```
t1_2 =

     0


t2_2 =

    0.1000


t3_2 =

    0.2000


t4_2 =

    0.3000


A_2 =

    1.0000    1.0000    1.0000    1.0000
    1.0000    0.9950    0.9900    0.9851
    1.0000    0.9801    0.9605    0.9414
    1.0000    0.9553    0.9127    0.8719


rref_A_2 =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1


detA_2 =

   6.5176e-11
```

```
t1_2 =

     0


t2_2 =

    0.2000


t3_2 =

    0.5000


t4_2 =

     1


A_2_check =

    1.0000    1.0000    1.0000    1.0000
    1.0000    0.9801    0.9605    0.9414
    1.0000    0.8776    0.7702    0.6759
    1.0000    0.5403    0.2919    0.1577


ans =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1


ans =

   1.7052e-05


t1_2 =

     0


t2_2 =

    0.1000


t3_2 =
```

```
    0.2000


A_2_f =

    1.0000         0    1.0000
    1.0000    0.0100    0.9900
    1.0000    0.0395    0.9605


ans =

    1.0000         0    1.0000
         0    1.0000   -1.0000
         0         0         0


ans =

   4.5189e-18


t1_2 =

     0


t2_2 =

     2


t3_2 =

     4


A_2_f =

    1.0000         0    1.0000
    1.0000    0.8268    0.1732
    1.0000    0.5728    0.4272


ans =

    1.0000         0    1.0000
         0    1.0000   -1.0000
         0         0         0


ans =
```

```
  -9.1796e-17
```

Question 3

```
format rat
% EXERCISE 54 FROM 4.5 IN "LINEAR ALGEBRA AND IT'S APPLICATIONS - 6tH
% EDITION:
%
% Let B = { l , cos t, cos2 t, ... , cos6 t} and C = { 1, cos t,
% cos 2t, . . . , cos 6t}. Assume the following trigonometric
% identities (see Exercise 45 in Section 4.1).
```

a)

B1_vec = 1; B2_vec = cos(t); B3_vec = cos(t)^2; B4_vec = cos(t)^3; B5_vec = cos(t)^4; B6_vec = cos(t)^5; B7_vec = cos(t)^6;

```
syms t
C0_3 = 1
C1_3 = cos(t)
C2_3 = -1 + 2*cos(t)^2
C3_3 = -3*cos(t) + 4*cos(t)^3
C4_3 = 1 - 8*cos(t)^2 + 8* cos(t)^4
C5_3 = 5*cos(t) - 20*cos(t)^3 + 16*cos(t)^5
C6_3 = -1 + 18*cos(t)^2 - 48*cos(t)^4 + 32*cos(t)^6

C0_3coords = [1;0;0;0;0;0;0]
C1_3coords = [0;1;0;0;0;0;0]
C2_3coords = [-1;0;2;0;0;0;0]
C3_3coords = [0;-3;0;4;0;0;0]
C4_3coords = [1;0;-8;0;8;0;0]
C5_3coords = [0;5;0;-20;0;16;0]
C6_3coords = [-1;0;18;0;-48;0;32]

C_augMat = [C0_3coords C1_3coords C2_3coords C3_3coords C4_3coords C5_3coords
 C6_3coords]
rref(C_augMat)

% C is a linearly independant of H because it's augmented matrix of
% B-coords of C has a pivot in every row (meaning it is linearly
% independent). C is isomorphic to B, so by converting C to B coordinate
% and showing that the set{ [C]B1,...,[C]B6 } is linearly independent to H, we
% show that the set {C1,...C6} is also linearly dependent to H (def of
% Isomorphism).
```

b)

C is a basis for H for the same reasons from a). To preface, lets say the set { [C]B1,...,[C]B6 } = the matix C_b. the basis Col(C_b) is every column, or Col(C_b) = {[C1}B,...[C]B6}, which means that all of C_b is a basis for H. Again, since C_b is isomorphic, that means that the set {C1,...C6} in the original coordinate space is also a basis for H

```
C0_3 =

       1

C1_3 =

cos(t)

C2_3 =

2*cos(t)^2 - 1

C3_3 =

4*cos(t)^3 - 3*cos(t)

C4_3 =

8*cos(t)^4 - 8*cos(t)^2 + 1

C5_3 =

5*cos(t) - 20*cos(t)^3 + 16*cos(t)^5

C6_3 =

18*cos(t)^2 - 48*cos(t)^4 + 32*cos(t)^6 - 1

C0_3coords =

       1
       0
       0
       0
       0
       0
       0

C1_3coords =

       0
       1
       0
       0
       0
       0
```

```
     0


C2_3coords =

    -1
     0
     2
     0
     0
     0
     0


C3_3coords =

     0
    -3
     0
     4
     0
     0
     0


C4_3coords =

     1
     0
    -8
     0
     8
     0
     0


C5_3coords =

     0
     5
     0
   -20
     0
    16
     0


C6_3coords =

    -1
     0
    18
     0
   -48
```

```
     0
    32


C_augMat =

  Columns 1 through 5

     1               0              -1               0               1
     0               1               0              -3               0
     0               0               2               0              -8
     0               0               0               4               0
     0               0               0               0               8
     0               0               0               0               0
     0               0               0               0               0

  Columns 6 through 7

     0              -1
     5               0
     0              18
   -20               0
     0             -48
    16               0
     0              32


ans =

  Columns 1 through 5

     1               0               0               0               0
     0               1               0               0               0
     0               0               1               0               0
     0               0               0               1               0
     0               0               0               0               1
     0               0               0               0               0
     0               0               0               0               0

  Columns 6 through 7

     0               0
     0               0
     0               0
     0               0
     0               0
     1               0
     0               1
```

Question 4

```
A_4 = [-2 4 1 8 2; 1 1 4 11 11; 1 -1 1 1 3; -2 6 4 18 10]
```

a)

```
rank(A_4)
```

b)

We can find the dimension of the Col, Null, and Row space by using the following relationships: dim(Nul(A)) => Rank(A)+dim(Nul(A)) = n, n-Rank(A) = dim(Nul(A)) = 5-2 = 3 dim(Col(A)) => Rank(A) = dim(Col(A)) = 2 dim(Row(A)) => Rank(A) = dim(Row(A)) = 2

c)

```
reducedA_4 = rref(A_4)

% Using the output from the above function, we will be defining the Row and
% Column space, find the basis for the following: Null, Row, and Col
%%%% ii)
basis_ColA_4 = [A_4(:,1) A_4(:,2)]

%%%% iii)
basis_RowA_4 = [reducedA_4(1,:); reducedA_4(2,:)]

%%%% i)
% The basis of Nul(A) is a little more involved
% basis of Nul(A) = Nul(A)
zero_vec = [0;0;0;0]
NulA_4 = [reducedA_4 zero_vec]
% x1 = -5/2x3 -6x4 -7x5
% x2 = -3/2x3 -5x4 - 4x5 -> we do see it though
% x3 = free
% x4 = free
% x5 = free
% x = x3[-5/2;-3/2;1;0;0] + x4[-6;-5;0;1;0] + x5[-7;-4;0;0;1]
x3_ph = [-5/2;-3/2;1;0;0]
x4_ph = [-6;-5;0;1;0]
x5_ph = [-7;-4;0;0;1]
basis_NulA_4 = [x3_ph x4_ph x5_ph]
```

```
A_4 =

    -2     4     1     8     2
     1     1     4    11    11
     1    -1     1     1     3
    -2     6     4    18    10


ans =

     2
```

```
reducedA_4 =

      1              0              5/2            6              7
      0              1              3/2            5              4
      0              0              0              0              0
      0              0              0              0              0


basis_ColA_4 =

     -2              4
      1              1
      1             -1
     -2              6


basis_RowA_4 =

      1              0              5/2            6              7
      0              1              3/2            5              4


zero_vec =

      0
      0
      0
      0


NulA_4 =

  Columns 1 through 5

      1              0              5/2            6              7
      0              1              3/2            5              4
      0              0              0              0              0
      0              0              0              0              0

  Column 6

      0
      0
      0
      0


x3_ph =

     -5/2
     -3/2
      1
      0
      0
```

```
x4_ph =

      -6
      -5
       0
       1
       0


x5_ph =

      -7
      -4
       0
       0
       1


basis_NulA_4 =

      -5/2          -6             -7
      -3/2          -5             -4
       1             0              0
       0             1              0
       0             0              1
```

Question 5

a)

```
v1_5 = [1;1;-1;2]
v2_5 = [3;5;0;5]
v3_5 = [1;3;2;1]
v4_5 = [5;8;1;-3]
v5_5 = [3;7;3;4]
```

b)

```
A_5 = [v1_5 v2_5 v3_5 v4_5 v5_5]
% the basis for the span of is the basis of the Col(A), so we find that
rref(A_5)
% using the result from the previous command, we can see that A_5 has
% pivots in columns 1, 2, and 4 so the basis for A is the basis for the
% Col(A) is just:
basis_A_equals_basis_ColA = [A_5(:,1) A_5(:,2) A_5(:,4)]
```

c)

for personal reference

```
P1_coords = v1_5
P2_coords = v2_5
P3_coords = v3_5
P4_coords = v4_5
P5_coords = v5_5

% span W =
% {1+t-t^2+2t^3, 3+5t+5t^3, 1+3t+2t^2+3t^3, 5+8t+t^2-3t^3, 3+7t+3t^2+4t^3}
% since the basis of P is col(1,2 and 4), and since W is isomorphic with P,
% then the basis of W should be col(1, 2,4) of the set:
% a basis of W = Col(W) = {1+t-t^2+2t^3, 3+5t+5t^3, 5+8t+t^2-3t^3}
% then dim(W) = 5 (if we're talking about the dimension of W itself?
% but the dim(W) in reference to the bases would be
% dim(col(W)) which is 3
```

d)

W != P3 because the basis of W only had a dimension of 3, where as P3 has a dimension of 4. This means that W can not equal P3, since they do not have the same dimensions, i.e. W is missing a dimension if it wanted to encompass all of P3. However, the elements of W still are in P3

```
v1_5 =

       1
       1
      -1
       2


v2_5 =

       3
       5
       0
       5


v3_5 =

       1
       3
       2
       1


v4_5 =

       5
       8
       1
      -3
```

```
v5_5 =

     3
     7
     3
     4


A_5 =

     1     3     1     5     3
     1     5     3     8     7
    -1     0     2     1     3
     2     5     1    -3     4


ans =

     1     0    -2     0    -3
     0     1     1     0     2
     0     0     0     1     0
     0     0     0     0     0


basis_A_equals_basis_ColA =

     1     3     5
     1     5     8
    -1     0     1
     2     5    -3


P1_coords =

     1
     1
    -1
     2


P2_coords =

     3
     5
     0
     5


P3_coords =

     1
     3
     2
```

```
     1


P4_coords =

     5
     8
     1
    -3


P5_coords =

     3
     7
     3
     4
```

Question 6

a)

```
A1coords_6 = [-12;-2;1;2;3;2]
A2coords_6 = [-4;1;0;0;-3;2]
A3coords_6 = [0;-1;2;4;5;-2]
A4coords_6 = [-3;2;2;4;-2;1]
```

b)

```
zero_vec_6 = [0;0;0;0;0;0]
augA_6 = [A1coords_6 A2coords_6 A3coords_6 A4coords_6 zero_vec_6]
rref(augA_6)

% The ouput generated from the previous command shows that when solving
% for the Augmented matrix Ax = 0 , we find that x4 is a free variable,
% meaning that there are infinite non-trivial linear combinations in which
% Ax = 0, showing that the set {A1-A4} is dependent. (Independence requires
% that the only solution to Ax = 0 is the trival solution)
% x1 = 1/2*x4
% x2 = -9/4*x^4
% x3 = -5/4*x^4
% x4 = free
% x5 = 0
```

c)

```
augA_6_again = [A1coords_6 A2coords_6 A3coords_6 A4coords_6]
rref(augA_6_again)

% From the output generated by the previous commands, we see that the
```

```
% linear combination for the coordinates of A1,A2,A3 = A4 is

% (-1/2)*A1coords + (9/4)*A2coords + (5/4)*A3coords = A4coords
A4_test = (-1/2)*A1coords_6 + (9/4)*A2coords_6 + (5/4)*A3coords_6
if (A4_test == A4coords_6)
    fprintf("The statement for the values of the linear combination of\n")
    fprintf("c1A1coord + c2A2coord + c3A3coord = A4coord is true!\n")
end

% this being the case, we can rewrite the matricies A1-A4 with the same
% weights:

% (-1/2)*A1 + (9/4)*A2 + (5/4)*A3 = A4
% and two prove this statement:
A1_6 = [-12 -2 1; 2 3 2]
A2_6 = [-4 1 0; 0 -3 2]
A3_6 = [0 -1 2; 4 5 -2]
A4_6 = [-3 2 2; 4 -2 1]
A4_test_2 = (-1/2)*A1_6 + (9/4)*A2_6 + (5/4)*A3_6

if (A4_test_2 == A4_6)
    fprintf("The same weights hold true for the Matrices A1-A4, meaning\n")
    fprintf("We can write them as a linear combination using the same \n")
    fprintf("weights as for the coordinates:\n")
    fprintf("c1A1 + c2A2 + c3A3 = A4\n")
end
```

```
A1coords_6 =

   -12
    -2
     1
     2
     3
     2


A2coords_6 =

    -4
     1
     0
     0
    -3
     2


A3coords_6 =

     0
    -1
     2
     4
```

```
     5
    -2


A4coords_6 =

    -3
     2
     2
     4
    -2
     1


zero_vec_6 =

     0
     0
     0
     0
     0
     0


augA_6 =

   -12              -4               0              -3               0
    -2               1              -1               2               0
     1               0               2               2               0
     2               0               4               4               0
     3              -3               5              -2               0
     2               2              -2               1               0


ans =

     1               0               0              -1/2             0
     0               1               0               9/4             0
     0               0               1               5/4             0
     0               0               0               0               0
     0               0               0               0               0
     0               0               0               0               0


augA_6_again =

   -12              -4               0              -3
    -2               1              -1               2
     1               0               2               2
     2               0               4               4
     3              -3               5              -2
     2               2              -2               1
```

```
ans =

       1              0              0             -1/2
       0              1              0              9/4
       0              0              1              5/4
       0              0              0              0
       0              0              0              0
       0              0              0              0


A4_test =

      -3
       2
       2
       4
      -2
       1
```

The statement for the values of the linear combination of c1A1coord + c2A2coord + c3A3coord = A4coord is true!

```
A1_6 =

     -12             -2              1
       2              3              2


A2_6 =

      -4              1              0
       0             -3              2


A3_6 =

       0             -1              2
       4              5             -2


A4_6 =

      -3              2              2
       4             -2              1


A4_test_2 =

      -3              2              2
       4             -2              1
```

The same weights hold true for the Matrices A1-A4, meaning We can write them as a linear combination using the same weights as for the coordinates:

```
c1A1 + c2A2 + c3A3 = A4
```

Published with MATLAB® R2022b