MATLAB PROJECT 3

GROUP # 27

1. **Jake Sanchez**
2. **Brandon Miguel**
3. **David Rowe**
4. **Nicolas Santiago**
5. **Charles Richardson**
6. **Nic Morita**
7. **Nic Morita** (OG)

Part I: Subspaces & Bases

Exercise 1

```
type columnspaces
```

```
function []=columnspaces(A,B)
m=size(A,1);
n=size(B,1);
if m~=n
    disp("Col A and Col B are subspaces of different spaces"); return
else
    fprintf('Col A and Col B are subspaces of R^%i\n',m); k=rank(A);
    fprintf('Col A dimensions:%d\n',k);
    l=rank(B);
    fprintf('Col B dimensions:%d\n',l);
    if l==m && k==m
        fprintf('Col A = Col B=R^%i\n',m);
        return
    else
        if k~=l
            disp("The dimensions of Col A and Col B are different");
        else
            if isequal(A,B)
                disp("Col A = Col B");
            else
                disp("The dimensions of Col A and Col B are the same, but Col A ~= Col B");
            end
       end
    end
end
end
```

```
format
```

Part a

```
A=[2 -4 -2 3;6 -9 -5 8;2 -7 -3 9;4 -2 -2 -1;-6 3 3 4]
```

```
A = 5×4
```

```
     2    -4    -2     3
     6    -9    -5     8
     2    -7    -3     9
     4    -2    -2    -1
    -6     3     3     4
```

```
B=rref(A)
```

```
B = 5×4
    1.0000         0   -0.3333         0
         0    1.0000    0.3333         0
         0         0         0    1.0000
         0         0         0         0
         0         0         0         0
```

```
columnspaces(A,B)
```

```
Col A and Col B are subspaces of R^5
Col A dimensions:3
Col B dimensions:3
The dimensions of Col A and Col B are the same, but Col A ~= Col B
```

Part b

```
A=[2 -4 -2 3;6 -9 -5 8;2 -7 -3 9;4 -2 -2 -1;-6 3 3 4];
B=([rref(A);zeros(5,4)])'
```

```
B = 4×10
    1.0000         0         0         0         0         0         0         0...
         0    1.0000         0         0         0         0         0         0
   -0.3333    0.3333         0         0         0         0         0         0
         0         0    1.0000         0         0         0         0         0
```

```
A=A'
```

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

```
columnspaces(A,B)
```

```
Col A and Col B are subspaces of R^4
Col A dimensions:3
Col B dimensions:3
The dimensions of Col A and Col B are the same, but Col A ~= Col B
```

Part c

```
A=magic(5)
```

```
A = 5×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

```
B=ones(5)
```

```
B = 5×5
```

```
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
```

```
columnspaces(A,B)
```

```
Col A and Col B are subspaces of R^5
Col A dimensions:5
Col B dimensions:1
The dimensions of Col A and Col B are different
```

Part d

```
A=magic(4)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
B=eye(4)
```

```
B = 4×4
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
columnspaces(A,B)
```

```
Col A and Col B are subspaces of R^4
Col A dimensions:3
Col B dimensions:4
The dimensions of Col A and Col B are different
```

Part e

```
A=magic(4)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
B=[eye(3);zeros(1,3)]
```

```
B = 4×3
     1     0     0
     0     1     0
     0     0     1
     0     0     0
```

```
columnspaces(A,B)
```

```
Col A and Col B are subspaces of R^4
Col A dimensions:3
Col B dimensions:3
The dimensions of Col A and Col B are the same, but Col A ~= Col B
```

Part f

```
A=magic(3)
```

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

```
B=[hilb(3),eye(3)]
```

```
B = 3×6
    1.0000    0.5000    0.3333    1.0000         0         0
    0.5000    0.3333    0.2500         0    1.0000         0
    0.3333    0.2500    0.2000         0         0    1.0000
```

```
columnspaces(A,B)
```

```
Col A and Col B are subspaces of R^3
Col A dimensions:3
Col B dimensions:3
Col A = Col B=R^3
```

Comment

The elementary row operations change the column space. since the elementary row operations do not affect the linear dependence relations among the columns. This means that the operations change the column space since linear dependence is unchaged.

Exercise 2

Part 1

```
type shrink
```

```
function B = shrink(A)
[~,pivot] = rref(A);
B = A(:,pivot);
end
```

```
A=magic(4);
A(:,3)=A(:,2)
```

```
A = 4×4
    16     2     2    13
     5    11    11     8
     9     7     7    12
     4    14    14     1
```

```
rref(A);
[R,pivot]=rref(A)
```

```
R = 4×4
     1     0     0     0
     0     1     1     0
     0     0     0     1
     0     0     0     0
pivot = 1×3
```

```
     1     2     4
```

Output: The rref basically shows us the **reduced row echelon form** of the given matrix that was alreadymanipulated being matrix A. With that in mind we are now trying to display "two matrices": One showingthe reduced form(as shown with R = 4x4) asigining it to 'R' and the other showing which columns are pivotcolumns(1, 2, 4).

```
B=A(:,pivot)
```

```
B = 4×3
    16     2    13
     5    11     8
     9     7    12
     4    14     1
```

Output: It would print out the manipulated matrix given at the beginning however it would only print out thecolumns that gives us a pivot position being that it only prints out columns 1, 2, and 4.

```
[~,pivot]=rref(A)
```

```
pivot = 1×3
     1     2     4
```

Part 2

```
A=[2 -4 -2 3;6 -9 -5 8;2 -7 -3 9;4 -2 -2 -1;-6 3 3 4]
```

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

```
B=shrink(A)
```

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

```
type columnspaces
```

```
function []=columnspaces(A,B)
m=size(A,1);
n=size(B,1);
if m~=n
    disp("Col A and Col B are subspaces of different spaces"); return
else
    fprintf('Col A and Col B are subspaces of R^%i\n',m); k=rank(A);
    fprintf('Col A dimensions:%d\n',k);
    l=rank(B);
    fprintf('Col B dimensions:%d\n',l);
    if l==m && k==m
        fprintf('Col A = Col B=R^%i\n',m);
        return
    else
        if k~=l
```

```
            disp("The dimensions of Col A and Col B are different");
        else
            if isequal(A,B)
                disp("Col A = Col B");
            else
                disp("The dimensions of Col A and Col B are the same, but Col A ~= Col B");
            end
       end
    end
end
end
```

```
columnspaces(A,B)
```

```
Col A and Col B are subspaces of R^5
Col A dimensions:3
Col B dimensions:3
The dimensions of Col A and Col B are the same, but Col A ~= Col B
```

```
%The set of the columns of B forms a basis for the column space of A since
%the 1st, 2nd, and 4th columns based off rref(A) are the only pivot columns.
A=[2 -4 -2 3;6 -9 -5 8;2 -7 -3 9;4 -2 -2 -1;-6 3 3 4]
```

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

```
R=rref((A'))
```

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

```
M=shrink(R')
```

```
M = 5×3
     1     0     0
     0     1     0
     0     0     1
     0     1    -1
    -2    -1     2
```

```
B=colspace(sym(A))
```

B =

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & -1 \\ -2 & -1 & 2 \end{pmatrix}$$

```
D = double(B)
```

```
D = 5×3
     1     0     0
```

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

```
isequal(D,M)
```

```
ans = logical
   1
```

```
columnspaces(A,B)
```

```
Col A and Col B are subspaces of R^5
Col A dimensions:3
Col B dimensions:3
The dimensions of Col A and Col B are the same, but Col A ~= Col B
```

Bonus 1

```
colspace(sym(A))
```

ans =

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & -1 \\ -2 & -1 & 2 \end{pmatrix}$$

Bonus 2

```
type columnspaces_1
```

```
function []=columnspaces_1(A,B)
m=size(A,1);
n=size(B,1);
if m~=n
    disp("Col A and Col B are subspaces of different spaces"); return
else
    fprintf('Col A and Col B are subspaces of R^%i\n',m); k=rank(A);
    fprintf('Col A dimensions:%d\n',k);
    l=rank(B);
    fprintf('Col B dimensions:%d\n',l);
    if l==m && k==m
        fprintf('Col A = Col B=R^%i\n',m);
        return
    else
        if k~=l
            disp("The dimensions of Col A and Col B are different");
        else
            if (colspace(sym(A)) == colspace(sym(B)))
                disp("Col A = Col B");
            else
                disp("The dimensions of Col A and Col B are the same, but Col A ~= Col B");
            end
       end
    end
end
end
```

Exercise 3

```
% Display 'basis' & 'shrink' functions
type basis
```

```
function [B,D] = basis(A)
m = size(A,1);
B = shrink(A);
fprintf('a basis for Col A is the set of the columns of\n')
disp(B);
if rank(B) == m
 fprintf('a basis for R^%i is D=B\n',m)
 D = B;
else
 B = [B eye(m)];
 D = shrink(B);
 if rank(D) == m
 fprintf('a basis for R^%i is\n',m)
 disp(D);
 else
 fprintf('something definitely went wrong!')
 end
end
```

```
type shrink
```

```
function B = shrink(A)
[~,pivot] = rref(A);
B = A(:,pivot);
end
```

Part a

```
% Run basis function on the following matrices:
A=[1 0;0 0;0 0;0 1]
```

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

```
[B,D]=basis(A);
```

```
a basis for Col A is the set of the columns of
     1     0
     0     0
     0     0
     0     1
a basis for R^4 is
     1     0     0     0
     0     0     1     0
     0     0     0     1
     0     1     0     0
```

Part b

```
A=[0 0;2 0;3 0;0 0]
```

```
A = 4×2
     0     0
```

```
     2     0
     3     0
     0     0
```

```
[B,D]=basis(A);
```

```
a basis for Col A is the set of the columns of
     0
     2
     3
     0
a basis for R^4 is
     0     1     0     0
     2     0     1     0
     3     0     0     0
     0     0     0     1
```

Part c

```
A=magic(4)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
[B,D]=basis(A);
```

```
a basis for Col A is the set of the columns of
    16     2     3
     5    11    10
     9     7     6
     4    14    15
a basis for R^4 is
    16     2     3     1
     5    11    10     0
     9     7     6     0
     4    14    15     0
```

Part d

```
A=magic(5)
```

```
A = 5×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

```
[B,D]=basis(A);
```

```
a basis for Col A is the set of the columns of
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
a basis for R^5 is D=B
```

Part e

```
A=ones(4)
```

```
A = 4×4
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

```
[B,D]=basis(A);
```

```
a basis for Col A is the set of the columns of
     1
     1
     1
     1
a basis for R^4 is
     1     1     0     0
     1     0     1     0
     1     0     0     1
     1     0     0     0
```

Part II: Isomorphism & Change of Basis

Exercise 4

```
type closetozeroroundoff
```

```
function B = closetozeroroundoff(A,p)
A(abs(A)<10^-p)=0;
B=A;
end
```

```
type polyspace
```

```
function P=polyspace(B,Q,r)
format
u = sym2poly(B(1));
n = length(u);
P = zeros(n);
for i = 1:n
    P(:,i) = transpose(sym2poly(B(i)));
end
P = closetozeroroundoff(P,7);
fprintf('matrix of E-coordinate vectors of polynomials in B is\n')
P
if (rank(P) == n)
    sprintf('the polynomials in B form a basis for the subspace of P%d',n-1)
else
    sprintf('the polynomials in B do not form a basis for the subspace of P%d',n-1)
    fprintf('the reduced echelon form of P is\n')
    A = rref(P)
    return;
end
fprintf('the B-coordinate vector of Q is\n')
E = sym2poly(Q);
closetozeroroundoff(E,7);
A = zeros(n,n+1);
for i = 1:n
    for j = 1:n
        A(i,j) = P(i,j);
```

```
    end
    A(i,n+1) = E(i);
end
L = rref(A);
y = L(:,n+1)
R = P * r;
fprintf('the polynomial whose B-coordinates form the vector r is\n')
R = poly2sym(R)
```

```
syms x
```

Part a

```
B=[x^3+3*x^2,10^(-8)*x^3+x,10^(-8)*x^3+4*x^2+x,x^3+x]
```

B =

$$\left(x^3+3\,x^2 \quad \frac{x^3}{100000000}+x \quad \frac{x^3}{100000000}+4\,x^2+x \quad x^3+x\right)$$

```
Q=10^(-8)*x^3-2*x^2+x-1
```

Q =

$$\frac{x^3}{100000000}-2\,x^2+x-1$$

```
r=[1;-3;2;4]
```

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

```
P=polyspace(B,Q,r);
```

```
matrix of E-coordinate vectors of polynomials in B is
P = 4×4
     1     0     0     1
     3     0     4     0
     0     1     1     1
     0     0     0     0
ans =
'the polynomials in B do not form a basis for the subspace of P3'
the reduced echelon form of P is
A = 4×4
    1.0000         0         0    1.0000
         0    1.0000         0    1.7500
         0         0    1.0000   -0.7500
         0         0         0         0
```

Part b

```
B=[x^3-1,10^(-8)*x^3+2*x^2,10^(-8)*x^3+x,x^3+x]
```

B =

$$\left(x^3-1 \quad \frac{x^3}{100000000}+2\,x^2 \quad \frac{x^3}{100000000}+x \quad x^3+x\right)$$

```
Q=10^(-8)*x^3-2*x^2+x-1
```

Q =

$$\frac{x^3}{100000000} - 2\,x^2 + x - 1$$

```
r=[1;-3;2;4]
```

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

```
P=polyspace(B,Q,r);
```

```
matrix of E-coordinate vectors of polynomials in B is
P = 4×4
     1     0     0     1
     0     2     0     0
     0     0     1     1
    -1     0     0     0
ans =
'the polynomials in B form a basis for the subspace of P3'
the B-coordinate vector of Q is
y = 4×1
    1.0000
   -1.0000
    2.0000
   -1.0000
the polynomial whose B-coordinates form the vector r is
```

R = $5\,x^3 - 6\,x^2 + 6\,x - 1$

Part c

```
B=[x^4+x^3+x^2+1,10^(-8)*x^4+x^3+x^2+x+1,10^(-8)*x^4+x^2+x+1,10^(-8)*x^4+x+1,10^(-8)*x^4+1]
```

B =

$$\left(x^4 + x^3 + x^2 + 1 \quad \frac{x^4}{100000000} + x^3 + x^2 + x + 1 \quad \frac{x^4}{100000000} + x^2 + x + 1 \quad \frac{x^4}{100000000} + x + 1 \quad \frac{x^4}{100000(} \right.$$

```
Q=x^4-2*x+3
```

Q = $x^4 - 2\,x + 3$

```
M=magic(5);r=M(:,1)
```

```
r = 5×1
    17
    23
     4
    10
    11
```

```
P=polyspace(B,Q,r);
```

```
matrix of E-coordinate vectors of polynomials in B is
P = 5×5
```

```
     1     0     0     0     0
     1     1     0     0     0
     1     1     1     0     0
     0     1     1     1     0
     1     1     1     1     1
ans = 
'the polynomials in B form a basis for the subspace of P4'
the B-coordinate vector of Q is
y = 5×1
     1
    -1
     0
    -1
     4
the polynomial whose B-coordinates form the vector r is
```

R = $17\,x^4 + 40\,x^3 + 44\,x^2 + 37\,x + 65$

Part III: Application to Calculus

Exercise 5

```
syms x
format long
```

Part a

```
fun=@(x) x.*tan(x) + x + 1
```

```
fun = function_handle with value:

    @(x)x.*tan(x)+x+1
```

```
a=0;b=1;
n=(1:10)';
T=reimsum(fun,a,b,n)
```

T = 10×4 table

	n	Left	Middle	Right
1	1	1.00000...	1.773151...	3.55740...
2	2	1.38657...	1.881266...	2.66527...
3	3	1.54665...	1.906168...	2.39912...
4	4	1.63392...	1.915497...	2.27327...
5	5	1.68877...	1.919945...	2.20025...
6	6	2.31931...	2.610207...	2.15264...
7	7	2.26204...	2.500411...	2.11918...
8	8	1.77470...	1.924869...	2.09438...
9	9	1.79111...	1.925541...	1.68000...
10	10	2.16009...	2.314070...	2.06009...

```
n=[1;5;10;100;1000;10000];
```

```
T=reimsum(fun,a,b,n)
```

T = 6×4 table

	n	Left	Middle	Right
1	1	1.00000...	1.773151...	3.55740...
2	5	1.68877...	1.919945...	2.20025...
3	10	2.16009...	2.314070...	2.06009...
4	100	1.91534...	1.928067...	1.90534...
5	1000	1.92681...	1.928088...	1.92581...
6	10000	1.92831...	1.928444...	1.92821...

```
Int=integral(fun,a,b)
```

```
Int =
   1.928088301365176
```

Part b

```
fun=@(x) x.^4 - 2*x - 2
```

```
fun = function_handle with value:

    @(x)x.^4-2*x-2
```

```
a=0;b=3;
n=(1:10)';
T=reimsum(fun,a,b,n)
```

T = 10×4 table

	n	Left	Middle	Right
1	1	-6.0000...	0.187500...	219
2	2	-2.9062...	23.91796...	1.09593750...
3	3	5.00000...	29.18750...	80
4	4	10.5058...	31.09643...	66.7558...
5	5	14.3270...	31.99133...	59.3270...
6	6	17.0937...	32.48046...	54.5937...
7	7	50.4639...	74.91101...	51.3211...
8	8	20.8011...	32.96891...	48.9261...
9	9	22.0987...	33.10108...	47.0987...
10	10	45.0591...	60.24251...	45.6591...

```
n=[1;5;10;100;1000;10000];
T=reimsum(fun,a,b,n)
```

T = 6×4 table

	n	Left	Middle	Right
1	1	-6.0000...	0.187500...	219
2	5	14.3270...	31.99133...	59.3270...
3	10	45.0591...	60.24251...	45.6591...
4	100	34.6730...	35.83401...	34.7330...
5	1000	33.4875...	33.59995...	33.4935...
6	10000	33.5887...	33.59999...	33.5893...

```
Int=integral(fun,a,b)
```

```
Int =
  33.600000000000001
```

Comment

The accuracy of the Riemann sum depends on what the slope of the graph looks like. Thus, the optimal method of partitioning is not a one-size-fits-all answer. However, it is indisputable that the more partitions there are, the more accurate the prediciton will be. Therefore, the best approximation is made by the Reimann sum that uses 10000 partitions.

Exercise 6

```
type polint
```

```
function I=polint(P)
format compact
syms x
u = sym2poly(P);
n = length(u);
for i=1:n

    u(i) = u(i)/(n+1-i);
end
u(end+1) = 3;
I=poly2sym(u);
end
```

```
format
syms x
P = 6*x^5+5*x^4+4*x^3+3*x^2+2*x+6
```

P = $6\,x^5 + 5\,x^4 + 4\,x^3 + 3\,x^2 + 2\,x + 6$

```
I=polint(P)
```

I = $x^6 + x^5 + x^4 + x^3 + x^2 + 6\,x + 3$

```
isequal(I,int(P)+3)
```

```
ans = logical

   1
```

```
P=x^5-2*x^3+3*x+5
```

P = $x^5 - 2\,x^3 + 3\,x + 5$

```
I=polint(P)
```

I =

$$\frac{x^6}{6} - \frac{x^4}{2} + \frac{3\,x^2}{2} + 5\,x + 3$$

```
isequal(I,int(P)+3)
```

ans = *logical*

1

Part IV: Application to Markov Chains

Exercise 7

```
format
type markov
```

```
function q=markov(P,x0)
format
n=size(P,1);
sum=0;
for i=1:n
    for j=1:n
        sum = sum + P(j,i);
    end
    if (sum~=1)
        disp('P is not a stochastic matrix')
        q=[];
        return;
    end
    sum = 0;
end
disp('The steady-state vector of the system is:')
q=null(P-eye(n),'r');
for i=1:n
    sum = sum+q(i);
end
q=q/sum
k=0;
x=x0;
while(norm(x-q)>=10^(-7))
    x = P*x;
    k = k+1;
end
fprintf('The number of iterations is %i', k)
end
```

Part a

```
P=[.6 .3;.5 .7]
```

P = 2×2

```
    0.6000    0.3000
    0.5000    0.7000
```

```
x0=[.3;.7]
```

```
x0 = 2×1
    0.3000
    0.7000
```

```
q=markov(P,x0);
```

```
P is not a stochastic matrix
```

Part b

```
P=[.5 .3;.5 .7]
```

```
P = 2×2
    0.5000    0.3000
    0.5000    0.7000
```

```
q=markov(P,x0);
```

```
The steady-state vector of the system is:
q = 2×1
    0.3750
    0.6250
The number of iterations is 9
```

Part c

```
P=[.9 .2;.1 .8]
```

```
P = 2×2
    0.9000    0.2000
    0.1000    0.8000
```

```
x0=[.10;.90]
```

```
x0 = 2×1
    0.1000
    0.9000
```

```
q=markov(P,x0);
```

```
The steady-state vector of the system is:
q = 2×1
    0.6667
    0.3333
The number of iterations is 45
```

Part d

```
x0=[.81;.19]
```

```
x0 = 2×1
    0.8100
    0.1900
```

```
q=markov(P,x0);
```

```
The steady-state vector of the system is:
q = 2×1
    0.6667
    0.3333
The number of iterations is 41
```

Part e

```
P=[.90 .01 .09;.01 .90 .01;.09 .09 .90]
```

```
P = 3×3
    0.9000    0.0100    0.0900
    0.0100    0.9000    0.0100
    0.0900    0.0900    0.9000
```

```
x0=[.5; .3; .2]
```

```
x0 = 3×1
    0.5000
    0.3000
    0.2000
```

```
q=markov(P,x0);
```

```
The steady-state vector of the system is:
q = 3×1
    0.4354
    0.0909
    0.4737
The number of iterations is 128
```

Comment

The choice of the initial vector x0 does not change the steady-state vector q because q is unique. However, it does effect the number of iterations k because an x0 that is farther away from the steady-state vector q will take more iterations to converge to q.