Please include this page in your Group file, as a front page. Type in the group number and the names of all members WHO PARTICIPATED in the project.

Group #22

FIRST & LAST

1. Kevin Lemos(Leader)
2. Johnny Li
3. Hannah McEarchern
4. Matt Leonard

By signing your names above, each of you have confirmed that you did the work and agree with the work you submitted.

%Part 1

%Exercise 1

type subspace

function [] = subspace(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);

end

k = rank(A);

p = rank(B);

fprintf('The dimension of Col A is k=%i and the dimension of Col B is l=%i\n',k,p);

C = [A,B];

if k==p

if k == rank(C)

fprintf('Col A = Col B\n')

else

fprintf('k=p, the dimensions of Col A and Col B are the same, but Col A ~= Col B\n');

end

if k == m

fprintf('k = p = m, (%i=%i=%i) Col A is all R^%i, as is Col B\n',k,p,m,m);

return;

else

fprintf('Neither Col A nor Col B is all R^%i\n',m);

end

end

if k~=p

fprintf('k ~= p, the dimensions of Col A and Col B are different\n');

if k == m

fprintf('k=m, (%i=%i) Col A is all R^%i, Col B is not\n', k, m, m);

return;

else

if p == m

fprintf('p=m, (%i=%i) Col B is all R^%i, Col A is not', p, m, m);

return;

else

fprintf('Neither Col A nor Col B is all R^%i\n',m);

return;

end

end

end

%a)

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

A =

2 -4 -2 3

6 -9 -5 8

2 -7 -3 9

4 -2 -2 -1

-6 3 3 4

B=rref(A)

B =

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

\_

subspace(A,B)

Col A and Col B are subspaces of R^5

The dimension of Col A is k=3 and the dimension of Col B is p=3

k=p, the dimensions of Col A and Col B are the same, but Col A ~= Col B

Neither Col A nor Col B is all R^5

%b)

A=magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

B=eye(4)

B =

1 0 0 0

0 1 0 0

0 0 1 0

0 0 0 1

subspace(A,B)

Col A and Col B are subspaces of R^4

The dimension of Col A is k=3 and the dimension of Col B is p=4

k ~= p, the dimensions of Col A and Col B are different

p=m, (4=4) Col B is all R^4, Col A is not

%c)

A=magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

B=eye(3)

B =

1 0 0

0 1 0

0 0 1

subspace(A,B)

Col A and Col B are subspaces of different spaces

%d)

A=magic(5)

A =

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=eye(5)

B =

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

subspace(A,B)

Col A and Col B are subspaces of R^5

The dimension of Col A is k=5 and the dimension of Col B is p=5

Col A = Col B

k = p = m, (5=5=5) Col A is all R^5, as is Col B

%Elementary row operations won't change the dimensions, but the space's values will be changed.

%Exercise 2

type shrink

function B = shrink(A)

[~,pivot] = rref(A);

B = A(:,pivot);

>> type basis

function B = basis(A)

m=size(A,1);

A=shrink(A);

fprintf('a basis for Col A is \n');

B=A;

r = rank(B);

if r == m

fprintf('a basis for R^%i is \n',m);

else

B = [B eye(m)];

D = shrink(B);

if rank(D) == m

fprintf('a basis for R^%i is \n',m);

B=D;

else

disp ('What? It is not a basis!?');

end

end

A = magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

[~,pivot]=rref(A)

pivot =

1 2 3

%indices of pivot columns for A

B=A(:,pivot)

B =

16 2 3

5 11 10

9 7 6

4 14 15

%prints only the pivot columns

%a)

A = [1 0;0 0;0 0;0 1]

A =

1 0

0 0

0 0

0 1

B=basis(A)

a basis for Col A is

a basis for R^4 is

B =

1 0 0 0

0 0 1 0

0 0 0 1

0 1 0 0

%b)

A = [2 0;4 0;1 0;0 0]

A =

2 0

4 0

1 0

0 0

B = basis(A)

a basis for Col A is

a basis for R^4 is

B =

2 1 0 0

4 0 1 0

1 0 0 0

0 0 0 1

%c)

A=magic(3)

A =

8 1 6

3 5 7

4 9 2

B=basis(A)

a basis for Col A is

a basis for R^3 is

B =

8 1 6

3 5 7

4 9 2

%d)

A=magic(6)

A =

35 1 6 26 19 24

3 32 7 21 23 25

31 9 2 22 27 20

8 28 33 17 10 15

30 5 34 12 14 16

4 36 29 13 18 11

B=basis(A)

a basis for Col A is

a basis for R^6 is

B =

35 1 6 26 19 1

3 32 7 21 23 0

31 9 2 22 27 0

8 28 33 17 10 0

30 5 34 12 14 0

4 36 29 13 18 0

%Exercise 3

format compact

type closetozeroroundoff

function B = closetozeroroundoff( A )

%CLOSETOZEROROUNDOFF Summary of this function goes here

% Detailed explanation goes here

[m,n] = size(A);

for i=1:m

for j=1:n

if abs(A(i,j)) < 10^(-7)

A(i,j) = 0;

end

end

end

B = A;

end

type polyspace

function P = polyspace( B, Q, r )

% B is a vector of polynomials from vector space P, Q is a single

% polynomial from the same space and r is a vector with n components.

%

format rat,

u = sym2poly(B(1));

n = length(u);

C = zeros(n);

for i=1:n

C(:,i) = transpose(sym2poly(B(i)));

end

P = closetozeroroundoff(C);

if (rank(P) ~= n)

str = sprintf('The polynomials in B do not form a basis for P%d\n', n-1);

fprintf(str);

fprintf('The reduced echelon form of P is \n');

A = rref(P);

return

elseif rank(P) == n

fprintf ('The polynomials in B form a basis for the corresponding space\n');

fprintf ('The coordinates of the polynomial Q with respect to the basis P are \n');

T = transpose(sym2poly(Q));

y = inv(P) \* T;

y = closetozeroroundoff(y);

disp(y);

fprintf 'The coordinates of vector q of polynomial R with respect to basis B are \n';

q = P \* r;

q = transpose(closetozeroroundoff(q));

R = poly2sym(q);

disp(q);

disp(R);

end

syms x

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

B =

[ x^3 + 3\*x^2, x^3/100000000 + x, x^3/100000000 + 4\*x^2 + x, x^3 + x]

Q=10^(-8)\*x^3+x^2+6\*x

Q =

x^3/100000000 + x^2 + 6\*x

r=[2;-3;1;0]

r =

2

-3

1

0

P= polyspace(B,Q,r)

The polynomials in B do not form a basis for P3

The reduced echelon form of P is

P =

1 0 0 1

3 0 4 0

0 1 1 1

0 0 0 0

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

B =

[ x^3 - 1, x^3/100000000 + 2\*x^2, x^3/100000000 + x, x^3 + x]

P= polyspace(B,Q,r)

The polynomials in B form a basis for the corresponding space

The coordinates of the polynomial Q with respect to the basis P are

0

1/2

6

0

The coordinates of vector q of polynomial R with respect to basis B are

2 -6 1 -2

2\*x^3 - 6\*x^2 + x - 2

P =

1 0 0 1

0 2 0 0

0 0 1 1

-1 0 0 0

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],Q=x^4-1, r=diag(magic(5))

B =

[ x^4 + x^3 + x^2 + 1, x^4/100000000 + x^3 + x^2 + x + 1, x^4/100000000 + x^2 + x + 1, x^4/100000000 + x + 1, x^4/100000000 + 1]

Q =

x^4 - 1

r =

17

5

13

21

9

P= polyspace(B,Q,r)

The polynomials in B form a basis for the corresponding space

The coordinates of the polynomial Q with respect to the basis P are

1

-1

0

1

-2

The coordinates of vector q of polynomial R with respect to basis B are

Columns 1 through 4

17 22 35 39

Column 5

65

17\*x^4 + 22\*x^3 + 35\*x^2 + 39\*x + 65

P =

Columns 1 through 4

1 0 0 0

1 1 0 0

1 1 1 0

0 1 1 1

1 1 1 1

Column 5

0

0

0

0

1

diary off

diary on

format compact

%Part III Application to Calculus

% Exercise#4

>> type reimsum

function [T, I] = reimsum(P,a,b,n)

%Function to approximate the value of the definite integral of a

%polynomial using Riemann sums.

%Setup

N= length(n); %jth entry on n

c= zeros(1,N); %Initalize value of c to be zero

d= zeros(1,N); %Initalize value of d to be zero

f= zeros(1,N); %Initalize value of f to be zero

poly = sym2poly(P); %Transfer symbolic value to polynomial values

%Nested loop to calculate and set values

for j=1:N

h = (b - a)/n(j); %Riemann sum calculation

%Reset

temp1= 0; %temporary value holder

temp2= 0;

temp3= 0;

for i=1:n(j)

test1=a+i\*h; %value of left interval

temp1=temp1+polyval(poly,test1); %store

test2=a+(i-1)\*h; %value of middle interval

temp2=temp2+polyval(poly,test2); %store

test3=0.5\*(test1+test2); %value of left interval

temp3=temp3+polyval(poly,test3); %store

end

%Store

temp1=h\*temp1;

temp2=h\*temp2;

temp3=h\*temp3;

%Set value at the table position

c(j)=temp2;

d(j)=temp3;

f(j)=temp1;

end

d= closetozeroroundoff (d); %Round d

I = double(int(P, a, b)); %The value of the integral I

A = [n', c', d', f']; %Output vectors

T = array2table(A,'VariableNames',{'n','Left','Middle','Right'}); %Table

end

%Case a1 with [a, b]: [-1, 1] and n: n = [1:10]

>> a=-1;b=1;

>> n=[1:10]

n =

1 2 3 4 5 6 7 8 9 10

>> syms x

>> P=2\*x^4+4\*x^2-1

P =

2\*x^4 + 4\*x^2 - 1

>> [T,I]=reimsum(P,a,b,n)

T =

10×4 table

n Left Middle Right

\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_

1 10 -2 10

2 4 0.25 4

3 2.6255 0.89712 2.6255

4 2.125 1.1406 2.125

5 1.8899 1.2563 1.8899

6 1.7613 1.32 1.7613

7 1.6835 1.3586 1.6835

8 1.6328 1.3838 1.6328

9 1.598 1.4011 1.598

10 1.5731 1.4135 1.5731

I =

1.4667

%The 'int' function represented by I is a better approximation for

%the integral, thou the reimsum function is close to the value.

%Case a2 with [a, b]: [-1, 1] and n: n = [1, 5, 10, 100, 1000, 10000]

>> n = [1, 5, 10, 100, 1000, 10000]

n =

1 5 10 100 1000 10000

>> [T,I]=reimsum(P,a,b,n)

T =

6×4 table

n Left Middle Right

\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_

1 10 -2 10

5 1.8899 1.2563 1.8899

10 1.5731 1.4135 1.5731

100 1.4677 1.4661 1.4677

1000 1.4667 1.4667 1.4667

10000 1.4667 1.4667 1.4667

I =

1.4667

%The 'int' function and reimsum function have the exact same value.

%Case a3 with [a, b]: [-10, 10] and n: n = [1:10]

>> n = [1:10]

n =

1 2 3 4 5 6 7 8 9 10

>> a=-10

a =

-10

>> b=10

b =

10

>> [T,I]=reimsum(P,a,b,n)

T =

10×4 table

n Left Middle Right

\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

1 4.0798e+05 -20 4.0798e+05

2 2.0398e+05 26980 2.0398e+05

3 1.3986e+05 55025 1.3986e+05

4 1.1548e+05 66543 1.1548e+05

5 1.0385e+05 72172 1.0385e+05

6 97445 75309 97445

7 93551 77228 93551

8 91011 78484 91011

9 89264 79350 89264

10 88012 79972 88012

I =

8.2647e+04

%The 'int' function represented by I is a better approximation for

%the integral.

%Case a4 with [a, b]: [-10, 10] and n: n = [1, 5, 10, 100, 1000, 10000]

>> n = [1, 5, 10, 100, 1000, 10000]

n =

1 5 10 100 1000 10000

>> [T,I]=reimsum(P,a,b,n)

T =

6×4 table

n Left Middle Right

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

1 4.0798e+05 -20 4.0798e+05

5 1.0385e+05 72172 1.0385e+05

10 88012 79972 88012

100 82701 82620 82701

1000 82647 82646 82647

10000 82647 82647 82647

I =

8.2647e+04

%The 'int' function and reimsum function have the relatively the same

%value.

%Case b1 with [a, b]: [-1, 1] and n: n = [1:10]

>>P=x^3-2\*x

P =

x^3 - 2\*x

>> [T,I]=reimsum(P,a,b,n)

T =

10×4 table

n Left Middle Right

\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

1 2 0 -2

2 1 0 -1

3 0.66667 1.4803e-16 -0.66667

4 0.5 0 -0.5

5 0.4 -1.7764e-16 -0.4

6 0.33333 0 -0.33333

7 0.28571 1.9032e-16 -0.28571

8 0.25 0 -0.25

9 0.22222 9.8686e-17 -0.22222

10 0.2 -1.3323e-16 -0.2

I =

0

%The 'int' function represented by I is a rounded estimate of

%the integral.

%Case b2 with [a, b]: [-1, 1] and n: n = [1, 5, 10, 100, 1000, 10000]

>> [T,I]=reimsum(P,a,b,n)

T =

6×4 table

n Left Middle Right

\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

1 2 0 -2

5 0.4 -1.7764e-16 -0.4

10 0.2 -1.3323e-16 -0.2

100 0.02 9.77e-17 -0.02

1000 0.002 -1.1191e-16 -0.002

10000 0.0002 -8.5576e-17 -0.0002

I =

0

%The 'int' function represented by I is a rounded estimate of

%the integral.

%Case b3 with [a, b]: [-10, 10] and n: n = [1:10]

>> [T,I]=reimsum(P,a,b,n)

T =

10×4 table

n Left Middle Right

\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_

1 -19600 0 19600

2 -9800 0 9800

3 -6533.3 7.5791e-13 6533.3

4 -4900 0 4900

5 -3920 0 3920

6 -3266.7 1.5158e-12 3266.7

7 -2800 3.2482e-13 2800

8 -2450 0 2450

9 -2177.8 1.0105e-12 2177.8

10 -1960 0 1960

I =

0

%The 'int' function represented by I is a rounded estimate of

%the integral

%Case b4 with [a, b]: [-10, 10] and n: n = [1, 5, 10, 100, 1000, 10000]

>> [T,I]=reimsum(P,a,b,n)

T =

6×4 table

n Left Middle Right

\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_

1 -19600 0 19600

5 -3920 0 3920

10 -1960 0 1960

100 -196 1.0914e-12 196

1000 -19.6 1.4097e-13 19.6

10000 -1.96 -3.1605e-13 1.96

I =

0

%The 'int' function represented by I is a rounded estimate of

%the integral

diary off

diary on

format compact

%Part III Application to Calculus

% Exercise#5

>> type polint

function B=polint(P)

%Function to calculate the indefinite integral of the polynomial

%assigning a value 5 to an arbitrary constant.

P=sym(P); %setup P

N=length(coeffs(P)); %value of n

syms x; %enable variable x

%Retrieve the coefficient and Degree

%C=coefficient

%D=variable and expontential

[C, D] = coeffs(P);

%Loop to access all values

for i=1:N

Power = length(D) - i; %Power of value

P(i)=(C(i)\*x^(Power+1))/(Power+1);

end

B = P; %assignment of constant value

end

%Case a)

>>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

>> B=polint(P)

B =

[ x^6 + x^5 + x^4 + x^3 + x^2 + 6\*x]

%Antiderivative of Polynomial

>> int(P)

ans =

x^6 + x^5 + x^4 + x^3 + x^2 + 6\*x

%Same output

%Case b)

>> P=x^6-x^4+3\*x^2+1

P =

x^6 - x^4 + 3\*x^2 + 1

>> B=polint(P)

B =

[ x^7/7 + -x^5/5 + x^3 + x]

%Antiderivative of Polynomial

>> int(P)

ans =

x^7/7 - x^5/5 + x^3 + x

%Same output

diary off

%exercise 6

diary on

format compact

%a

P = [0.6 0.3; 0.5 0.7]

P =

0.6000 0.3000

0.5000 0.7000

x0 = [.4; 0.6]

x0 =

0.4000

0.6000

markov(P, x0)

P is not a stochastic matrix

%b

P = [0.5 0.3; 0.5 0.7]

P =

0.5000 0.3000

0.5000 0.7000

markov(P, x0)

vector:

0.3750

0.6250

k:

8

ans =

0.3750

0.6250

%xk and vector q are the same

%c

P = [0.9 0.2; .1 .8]

P =

0.9000 0.2000

0.1000 0.8000

x0 = [.12; .88]

x0 =

0.1200

0.8800

markov(P, x0)

vector:

0.6667

0.3333

k:

45

ans =

0.6667

0.3333

%d

x0 = [.14; .86]

x0 =

0.1400

0.8600

markov(P, x0)

vector:

0.6667

0.3333

k:

45

ans =

0.6667

0.3333

x0 = [.86; .14]

x0 =

0.8600

0.1400

markov(P, x0)

vector:

0.6667

0.3333

k:

42

ans =

0.6667

0.3333

%the vector q are the same. Altering x0 does not change the steady-state vector. It does change the ammount of iterations k.

%e

P = [0.9 0.01 0.09; 0.01 0.9 0.01; 0.09 0.09 0.9]

P =

0.9000 0.0100 0.0900

0.0100 0.9000 0.0100

0.0900 0.0900 0.9000

x0 = [0.5; 0.3; 0.2 ]

x0 =

0.5000

0.3000

0.2000

markov(P, x0)

vector:

0.4354

0.0909

0.4737

k:

128

ans =

0.4354

0.0909

0.4737

diary off