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 this project.

GROUP # 22

FIRST & LAST NAMES (UFID numbers are NOT required):

1. Kevin Lemos (Leader)
2. Johnny Li
3. Hannah McEachern
4. Matt Leonard
5. Vinny Martinson

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

diary on

format compact

%Exercise 1

type ele1

function E1 = ele1( n,r,i,j )

%ELE1 creates the first elementary matrix

% The function starts with an nxn eye matrix and then replaces row j with

% row j + row i \* r.

E1 = eye(n);

E1(j, :) = E1(j,:) + (E1(i,:)\*r);

end

type ele2

function E2 = ele2(n,i,j)

%ELE2 creates the second elementary matrix

%   it creates an nxn eye matrix and switches the rows.

E2 = eye(n);

E2([j i],:) = E2([i j],:);

end

type ele3

function E3 = ele3( n, j, k )

%ELE3 creates the third elementary matrix

%   This function creates an nxn eye matrix and then multiplies row j by k.

E3 = eye(n);

    E3(:,j) = (E3(:,j) \* k);

end

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

format compact

format rat

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

A =

       2              1              3

       1              0              2

       2              3             -1

E2 = ele2(3,1,2)

E2 =

       0              1              0

       1              0              0

       0              0              1

A1 = E2 \* A

A1 =

       1              0              2

       2              1              3

       2              3             -1

E1 = ele1(3,-2,1,2)

E1 =

       1              0              0

      -2              1              0

       0              0              1

A2 = E1 \* A1

A2 =

       1              0              2

       0              1             -1

       2              3             -1

E1 = ele1(3,-2,1,3)

E1 =

       1              0              0

       0              1              0

      -2              0              1

A3 = E1 \* A2

A3 =

       1              0              2

       0              1             -1

       0              3             -5

E1 = ele1(3,-3,2,3)

E1 =

       1              0              0

       0              1              0

       0             -3              1

A4 = E1 \* A3

A4 =

       1              0              2

       0              1             -1

       0              0             -2

diary off

diary on

format compact

% Exercise#2

type inverses

function D=inverses(A)

%Pre-Setup

[rows, columns] = size(A);

D=[rows columns];

% Takes an n x n matrix A

if rows ~= columns

disp('Error: Not a square matrix!');

D=[];

return;

end

%Determine whether A is invertible using the function rank.

%Rank: provides an estimate of the number of linearly independent

%columns of a full matrix.

c=rank(A);

%Matrix A is inversible if all its column is indpendent.

if(c==columns)

%Take the inverse of matrix A.

%Set matrix A with an identity matrix.

A = [A eye(columns)];

%Row Reduce Echolen Form.

B = rref(A);

%Retrieve only the inverse matrix and display in Matrix D.

D = B(:,[columns+1:end]);

%If matrix A does not have every column independent.

%Output D=[ ]

else

disp('Matrix A is not invertible')

D=[ ];

end

end

A=[4 0 -7 -7; -6 1 11 9; 7 -5 10 19; -1 2 3 -1]

A =

4 0 -7 -7

-6 1 11 9

7 -5 10 19

-1 2 3 -1

inverses(A)

ans =

-19 -14 0 7

-549 -401 -2 196

267 195 1 -95

-278 -203 -1 99

A=[1 -3 2 -4; -3 9 -1 5; 2 -6 4 -3; -4 12 2 7]

A =

1 -3 2 -4

-3 9 -1 5

2 -6 4 -3

-4 12 2 7

inverses(A)

Matrix A is not invertible

ans =

[]

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

inverses(A)

ans =

-0.0049 0.0512 -0.0354 0.0012 0.0034

0.0431 -0.0373 -0.0046 0.0127 0.0015

-0.0303 0.0031 0.0031 0.0031 0.0364

0.0047 -0.0065 0.0108 0.0435 -0.0370

0.0028 0.0050 0.0415 -0.0450 0.0111

A = magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

inverses(A)

Matrix A is not invertible

ans =

[]

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

inv(A)

ans =

-0.0049 0.0512 -0.0354 0.0012 0.0034

0.0431 -0.0373 -0.0046 0.0127 0.0015

-0.0303 0.0031 0.0031 0.0031 0.0364

0.0047 -0.0065 0.0108 0.0435 -0.0370

0.0028 0.0050 0.0415 -0.0450 0.0111

A = magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

inv(A)

[\_Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 1.306145e-17.]\_

ans =

1.0e+14 \*

0.9382 2.8147 -2.8147 -0.9382

2.8147 8.4442 -8.4442 -2.8147

-2.8147 -8.4442 8.4442 2.8147

-0.9382 -2.8147 2.8147 0.9382

%Originally magic(4) has no inverse which was correct output by the

%inverses function. But the inv function led to a incorrect output of values.

%This is because inv performs the matrix inversion using floating-point

%computations, it is close to but not exactly equal to, the estimated inverse

%which does not exist.

diary off

diary on

format compact

%Part 3

%Exercise 3

%Matthew Leonard

type solvesys

function [C,N] = solvesys(A)

[n,n]=size(A);

b= fix(10\*rand(n, 1))

if det (A)~= 0

x1= A\b;

x2= inv(A)\*b;

x3= rref([A b]);

format long, C = [x1,x2,x3];

n1= norm(x1-x2);

n2= norm(x2-x3);

n3= norm(x3-x1);

format long, N = [n1,n2,n3];

else

C=[];

N=[];

disp ('The system is either inconsistent or the solution is not unique.')

end

end

%a)

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

[C,N] = solvesys(A)

b =

3

9

0

4

3

7

[\_> In <a href="matlab:matlab.internal.language.introspective.errorDocCallback('solvesys', 'M:\solvesys.m', 5)" style="font-weight:bold">solvesys</a> (<a href="matlab: opentoline('M:\solvesys.m',5,0)">line 5</a>)]\_

[\_Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =

4.800964e-18.]\_

[\_Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =

4.800964e-18.]\_

[\_> In <a href="matlab:matlab.internal.language.introspective.errorDocCallback('solvesys', 'M:\solvesys.m', 6)" style="font-weight:bold">solvesys</a> (<a href="matlab: opentoline('M:\solvesys.m',6,0)">line 6</a>)]\_

C =

1.0e+15 \*

Columns 1 through 4

-2.251799813685249 -2.251799813685250 0.000000000000001 0

-2.251799813685249 -2.251799813685250 0 0.000000000000001

1.125899906842624 1.125899906842625 0 0

2.251799813685249 2.251799813685249 0 0

2.251799813685249 2.251799813685249 0 0

-1.125899906842623 -1.125899906842623 0 0

Columns 5 through 8

0 0 0 -0.000000000000002

0 0 0 -0.000000000000002

0.000000000000001 0 0 0.000000000000001

0 0.000000000000001 0 0.000000000000002

0 0 0.000000000000001 0.000000000000002

0 0 0 0

Column 9

0

0

0

0

0

0.000000000000001

N =

1.0e+16 \*

0.000000000000000 1.263819510961739 1.263819510961739

%b)

A=eye(5)

A =

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

[C,N] = solvesys(A)

b =

7

1

4

4

6

C =

7 7 1 0 0 0 0 7

1 1 0 1 0 0 0 1

4 4 0 0 1 0 0 4

4 4 0 0 0 1 0 4

6 6 0 0 0 0 1 6

N =

0 23.392179680693115 23.392179680693115

%c)

A=randi(20,4,4)

A =

15 14 20 16

16 4 7 6

6 3 12 11

14 10 5 14

[C,N] = solvesys(A)

b =

8

9

5

1

C =

Columns 1 through 4

0.553771842591277 0.553771842591277 1.000000000000000 0

-0.443102713453616 -0.443102713453616 0 1.000000000000000

0.598522517403040 0.598522517403040 0 0

-0.379599374911209 -0.379599374911209 0 0

Columns 5 through 7

0 0 0.553772070626003

0 0 -0.443103448275862

1.000000000000000 0 0.598522167487685

0 1.000000000000000 -0.379603399433428

N =

0.000000000000000 2.297162716445022 2.297162716445023

%d)

A=magic(3)

A =

8 1 6

3 5 7

4 9 2

[C,N] = solvesys(A)

b =

1

2

8

C =

Columns 1 through 4

0.369444444444444 0.369444444444444 1.000000000000000 0

0.827777777777778 0.827777777777778 0 1.000000000000000

-0.463888888888889 -0.463888888888889 0 0

Columns 5 through 6

0 0.369444444444444

0 0.827777777777778

1.000000000000000 -0.463888888888889

N =

0.000000000000000 1.903787728144069 1.903787728144070

%e)

A=hilb(6)

A =

1.0000 0.5000 0.3333 0.2500 0.2000 0.1667

0.5000 0.3333 0.2500 0.2000 0.1667 0.1429

0.3333 0.2500 0.2000 0.1667 0.1429 0.1250

0.2500 0.2000 0.1667 0.1429 0.1250 0.1111

0.2000 0.1667 0.1429 0.1250 0.1111 0.1000

0.1667 0.1429 0.1250 0.1111 0.1000 0.0909

[C,N]=solvesys(A)

b =

8

9

1

9

6

0

C =

1.0e+06 \*

Columns 1 through 4

-0.024702000001226 -0.024702000001188 0.000001000000000 0

0.621180000037189 0.621180000035740 0 0.000001000000000

-3.831240000260855 -3.831240000249520 0 0

9.278640000694983 9.278640000662900 0 0

-9.673020000780385 -9.673020000742971 0 0

3.636864000311486 3.636864000296168 0 0

Columns 5 through 8

0 0 0 0

0 0 0 0

0.000001000000000 0 0 0

0 0.000001000000000 0 0

0 0 0.000001000000000 0

0 0 0 0.000001000000000

Column 9

-0.024702000000000

0.621180000000000

-3.831240000000000

9.278639999999999

-9.673019999999999

3.636864000000000

N =

1.0e+07 \*

0.000000000005286 3.532297813080138 3.532297813093042

%Part (b) yields an output that places the vector b in two columns before the eye and it places it in one column after the eye.

%Norm calculates the sum of each squared element in a vector and takes the square root of them. Since the elements of the hilb function are so small, taking the square of each element and square rooting them yields a larger value than the other functions.

diary off

diary on

format compact

type arevol

function D = arevol( B )

%AREVOL calculates the area or volume of a parallelogram or parallelopid

%respectivly

if length(B) == 3

V1 = [B(1,2) - B(1,1); B(2,2) - B(2,1)];

V2 = [B(1,3) - B(1,1); B(2,3) - B(2,1)];

final = [V1, V2];

D = abs(det(final));

D = closetozeroroundoff(D);

if D == 0

disp("“The points lie on the same line and no parallelogram can be built");

else

disp("The area of the parallelogram is " + D);

end

else

V1=[B(1,2) - B(1,1); B(2,2) - B(2,1); B(3,2) - B(3,1)];

V2=[B(1,3) - B(1,1); B(2,3) - B(2,1); B(3,3) - B(3,1)];

V3=[B(1,4) - B(1,1); B(2,4) - B(2,1); B(3,4) - B(3,1)];

final = [V1, V2, V3];

D = abs(det(final));

D = closetozeroroundoff(D);

if D == 0

disp("“The points lie on the same line and no parallelpepid can be built");

else

disp("The area of the parallelpepid is " + D);

end

end

end

type closetozeroroundoff

function B=closetozeroroundoff(A)

[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;

B = randi([-10,10], 2, 3)

B =

4 9 -1

-4 -10 -2

D = arevol(B)

The area of the parallelogram is 20

D =

20

B = randi([-10,10], 3, 4)

B =

6 0 4 4

6 -1 5 3

-7 3 -5 -7

D = arevol(B)

The area of the parallelpepid is 32

D =

32

X = randi([-10,10], 2, 1)

X =

-8

0

B = [X,-X, 2\*X]

B =

-8 8 -16

0 0 0

D = arevol(B)

The points lie on the same line and no parallelogram can be built

D =

0

X = randi([-10,10], 3, 1), Y = randi([-10,10], 3, 1), B = [X, Y, X+Y, X-Y]

X =

10

-3

2

Y =

-6

5

-5

B =

10 -6 4 16

-3 5 2 -8

2 -5 -3 7

D = arevol(B)

The points lie on the same line and no parallelpepid can be built

D =

0

diary close

diary on

%Exercise 5

R1=[1 0; 0 -1]

R1 =

1 0

0 -1

%Reflect over X-axis.

R1=[0 1; 1 0]

R1 =

0 1

1 0

hold

Current plot held

grid

%Limit of axis size.

v=[-3 3 -3 3];

%Set axis limits value of v.

axis(v)

plot(R1)

R2=[-1 0 ; 0 -1]

R2 =

-1 0

0 -1

%Reflect over Y-axis.

plot(R2)

VS=[1 2; 0 1]

VS =

1 2

0 1

%Vertial Shear by 2.

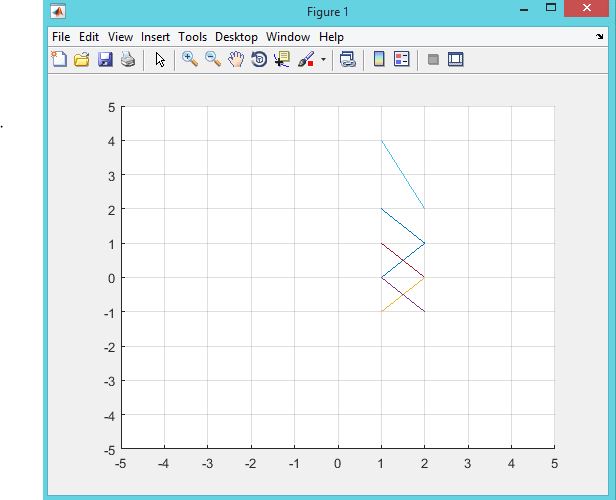
VS=[1 4; 0 2]

VS =

1 4

0 2

plot(VS)



type transf

function C = transf(A,E)

%Function to transform fucntion.

%The value of E is multiple by A.

E=A\*E;

%X is equal to the value of the 1 row, addition of all columns.

%X is equal to the value of the 2 row, addition of all columns.

x=E(1,:); y=E(2,:);

%Plot the system.

plot(x,y)

%Limit of axis size.

v=[-5 5 -5 5];

%Set axis limits value of v.

axis(v)

%Display axes grid lines.

grid

%The value of C equals to E.

C=E;

%Hide axes grid lines.

grid

end

E=[0 1 1 0 0; 0 0 1 1 0]

E =

0 1 1 0 0

0 0 1 1 0

A = eye(2);

hold

Current plot held

grid

C = transf(A,E)

C =

0 1 1 0 0

0 0 1 1 0

E = C;

A=VS

A =

1 2

0 1

C = transf(A,E)

C =

0 1 3 2 0

0 0 1 1 0

E = C;

A=R2

A =

-1 0

0 -1

C = transf(A,E)

C =

0 -1 -3 -2 0

0 0 -1 -1 0

A=R1

A =

1 0

0 -1

C = transf(A,E)

C =

0 1 3 2 0

0 0 -1 -1 0

E = C;

A=R2

A =

-1 0

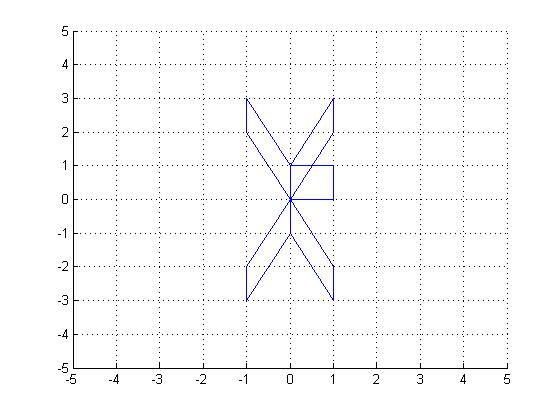
0 -1

C = transf(A,E)

C =

0 -1 -3 -2 0

0 0 1 1 0



diary on

% exercise 6

format compact

type cofactor

function C = cofactor( a)

%COFACTOR returns vector whose entries are the cofactors to Matrix a

N = length(a);

C = zeros(N);

for i=1:N;

for j =1:N;

temp = a;

temp(i, :) = [];

temp(:, j) = [];

C(i, j) = (-1)^(i+j) \* det(temp);

end

end

end

type determine

function D = determine( a, C )

%DETERMINE finds determinant

N = length(a);

D1 = zeros(N, 1);

D2 = zeros(N, 1);

for i=1:N

for j=1:N

D1(i) = D1(i) + (a(i,j) \* C(i,j));

end

end

for j=1:N

for i=1:N

D2(j) = D2(j) + (a(i,j) \* C(i,j));

end

end

good = true;

iter = 1;

while(good && iter<N)

for k=1:N

good = (D1(iter) - D2(k)) < 10^(-7);

end

iter = iter +1;

end

if good

D = D1(1);

else

disp("There is a problem with my code!");

D = [];

end

type inverse

function B = inverse( a, C, D )

%INVERSE Finds the inverse of a matrix

%we check to see if the matrix is invertable, and also make sure its square

%we use rank(a) to ensure that a has a pivot position at every column. det(A) can be slightly off, especially at 0.

if(rank(a) == length(a) &&length(a(1,:)) == length(a(:,1)))

B = (1/D) \* transpose(C);

else

disp("Matrix is not invertible!");

B = [];

end

format rat

a = diag([1,2,3,4])

a =

1 0 0 0

0 2 0 0

0 0 3 0

0 0 0 4

C = cofactor(a)

C =

24 0 0 0

0 12 0 0

0 0 8 0

0 0 0 6

D = determine(a, C)

D =

24

det(a)

ans =

24

%no difference

B = inverse(a, C, D)

B =

1 0 0 0

0 1/2 0 0

0 0 1/3 0

0 0 0 1/4

inv(a)

ans =

1 0 0 0

0 1/2 0 0

0 0 1/3 0

0 0 0 1/4

%no difference

a = ones(5)

a =

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

C = cofactor(a)

C =

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

D = determine(a, C)

D =

0

det(a)

ans =

0

%no difference

B = inverse(a, C, D)

Matrix is not invertible!

B =

[]

inv(a)

[\_Warning: Matrix is singular to working precision.]\_

ans =

1/0 1/0 1/0 1/0 1/0

1/0 1/0 1/0 1/0 1/0

1/0 1/0 1/0 1/0 1/0

1/0 1/0 1/0 1/0 1/0

1/0 1/0 1/0 1/0 1/0

%answers are not the same but they both suggest that the matrix is not inversible

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

C = cofactor(a)

C =

-25025 218725 -153400 23725 13975

259350 -189150 15600 -33150 25350

-179400 -23400 15600 54600 210600

5850 64350 15600 220350 -228150

17225 7475 184600 -187525 56225

D = determine(a, C)

D =

5070000

det(a)

ans =

5070000

%no difference

B = inverse(a, C, D)

B =

-77/15600 133/2600 -23/650 3/2600 53/15600

89/2063 -97/2600 -3/650 33/2600 23/15600

-59/1950 1/325 1/325 1/325 71/1950

73/15600 -17/2600 7/650 113/2600 -577/15600

43/15600 1/200 27/650 -9/200 98/8837

inv(a)

ans =

-77/15600 133/2600 -23/650 3/2600 53/15600

89/2063 -97/2600 -3/650 33/2600 23/15600

-59/1950 1/325 1/325 1/325 71/1950

73/15600 -17/2600 7/650 113/2600 -577/15600

43/15600 1/200 27/650 -9/200 98/8837

%no difference

a = magic(4)

a =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

C = cofactor(a)

C =

-136 -408 408 136

-408 -1224 1224 408

408 1224 -1224 -408

136 408 -408 -136

D = determine(a, C)

D =

1/4398046511104

det(a)

ans =

1/1947923714261

%not exactly the same, this is becouse of a rounding error as both numbers are essentially zero by MATLAB standards

B = inverse(a, C, D)

Matrix is not invertible!

B =

[]

inv(a)

[\_Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 4.625929e-18.]\_

ans =

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

%the results are diffenet but they both imply that the matrix is not invertible

a = hilb(4)

a =

1 1/2 1/3 1/4

1/2 1/3 1/4 1/5

1/3 1/4 1/5 1/6

1/4 1/5 1/6 1/7

C = cofactor(a)

C =

1/378000 -1/50400 1/25200 -1/43200

-1/50400 1/5040 -1/2240 1/3600

1/25200 -1/2240 3/2800 -1/1440

-1/43200 1/3600 -1/1440 1/2160

D = determine(a, C)

D =

1/6048000

det(a)

ans =

1/6048000

%no difference

B = inverse(a, C, D)

B =

16 -120 240 -140

-120 1200 -2700 1680

240 -2700 6480 -4200

-140 1680 -4200 2800

inv(a)

ans =

16 -120 240 -140

-120 1200 -2700 1680

240 -2700 6480 -4200

-140 1680 -4200 2800

%no difference

diary close