MATLAB PROJECT 2

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 # \_\_\_\_3\_\_\_\_\_\_

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

\_1.\_\_\_\_\_\_Kevin Nguyen\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_2.\_\_\_Chamara Gunaratne\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ­­­

\_3.\_\_\_\_\_Nicolas Miller\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_4.\_\_\_\_\_\_Carl Lin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_5.\_\_\_\_\_\_Cayle Gao\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_6.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

%Exercise 1

type ele1

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

%This function does row replacement

E1=eye(n);

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

end

type ele2

function E2=ele2(n,i,j)

%This function interchanges rows

E2=eye(n);

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

end

type ele3

function E3=ele3(n,j,k)

%This function multiples rows by a scalar

E3=eye(n);

E3(j,:)=k\*E3(j,:);

end

E1=ele1(4,3,2,4)

E1 =

1 0 0 0

0 1 0 0

0 0 1 0

0 3 0 1

%This matrix was created by multiplying row 2 by 3 and then replacing row 4 with the sum of rows 2 and 4

E2=ele2(4,2,4)

E2 =

1 0 0 0

0 0 0 1

0 0 1 0

0 1 0 0

%This matrix was produced by switching rows 2 and 4

E3=ele3(4,4,5)

E3 =

1 0 0 0

0 1 0 0

0 0 1 0

0 0 0 5

%This matrix was produced by multiplying the fourth row of the matrix by 5

det(E1)

ans =

1

%This determinant is the same as the determinant of the orignal identity matrix because row replacement does not change the determinant

det(E2)

ans =

-1

%This determinant is the opposite sign of the determinant of the original identity matrix because the rows of the matrix were interchanged once so the determinant was multiplied by -1 one time

det(E3)

ans =

5

%This determinant is 5 times the original determinant because one of the rows was multiplied by 5

M=magic(4)

M =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

M\*E1

ans =

16 41 3 13

5 35 10 8

9 43 6 12

4 17 15 1

%This matrix was obtained by replacing the entries in the second column of M by the sum of the second entry and 3 times the fourth entry of each row in matrix M.

%This is because the second column in E1 had a 3 in the fourth row

M\*E2

ans =

16 13 3 2

5 8 10 11

9 12 6 7

4 1 15 14

%The second and fourth columns of matrix M have been switched because the second and fourth rows of E2 were switched.

M\*E3

ans =

16 2 3 65

5 11 10 40

9 7 6 60

4 14 15 5

%The last column of M has been multiplied by 5 because the last row of E3 was multiplied by 5.

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

E1=ele1(5,3,4,2)

E1 =

1 0 0 0 0

0 1 0 3 0

0 0 1 0 0

0 0 0 1 0

0 0 0 0 1

E2=ele2(5,1,3)

E2 =

0 0 1 0 0

0 1 0 0 0

1 0 0 0 0

0 0 0 1 0

0 0 0 0 1

E3=ele3(5,5,2)

E3 =

1 0 0 0 0

0 1 0 0 0

0 0 1 0 0

0 0 0 1 0

0 0 0 0 2

A=E3\*E2\*E1\*A

A =

0 0 1 0 0

0 1 0 3 0

1 0 0 0 0

0 0 0 1 0

0 0 0 0 2

%Since the original identity matrix A was invertible the obtained matrix A is also invertible because the row operations did not make the determinant of the matrix 0.

%The new determinant of matrix A is -2 because a row was scaled by 2 and two rows were interchanged. Therefore, the matrix is invertable.

C=inv(A)

C =

0 0 1.0000 0 0

0 1.0000 0 -3.0000 0

1.0000 0 0 0 0

0 0 0 1.0000 0

0 0 0 0 0.5000

C=inv(E3)\*inv(E2)\*inv(E1)

C =

0 0 1.0000 0 0

0 1.0000 0 -3.0000 0

1.0000 0 0 0 0

0 0 0 1.0000 0

0 0 0 0 0.5000

%Exercise 2

type Sudoku

function S=sudoku(A)

[m,n]=size(A);

if m==n

S1=sum(A)

S2=sum(A,2)

epsilon = 10^-7;

if (all(diff(S1)<epsilon) && all(diff(S2,2)<epsilon) && (all((S1 - transpose(S2)) < epsilon)))

S=S1(1,1);

return

else

S=sym('undefined');

disp('ERROR using sudoku: the sums are different')

return

end

else

disp('ERROR using sudoku: the matrix is not square')

S=sym('undefined');

end

end

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

S=sudoku(A)

S1 =

65 65 65 65 65

S2 =

65

65

65

65

65

S =

65

A=diag(diag(magic(6)))

A =

35 0 0 0 0 0

0 32 0 0 0 0

0 0 2 0 0 0

0 0 0 17 0 0

0 0 0 0 14 0

0 0 0 0 0 11

S=sudoku(A)

S1 =

35 32 2 17 14 11

S2 =

35

32

2

17

14

11

ERROR using sudoku: the sums are different

S =

undefined

A=ones(4,5)

A =

1 1 1 1 1

1 1 1 1 1

1 1 1 1 1

1 1 1 1 1

S=sudoku(A)

ERROR using sudoku: the matrix is not square

S =

undefined

A=randi(10,3,3)

A =

5 10 9

10 7 10

8 1 7

S=sudoku(A)

S1 =

23 18 26

S2 =

24

27

16

ERROR using sudoku: the sums are different

S =

undefined

B=randi(10,4,4); C=inv(B); A=B\*C

A =

1.0000 -0.0000 0 0

0 1.0000 0 0

0.0000 -0.0000 1.0000 0

0 0 0 1.0000

S=sudoku(A)

S1 =

1.0000 1.0000 1.0000 1.0000

S2 =

1.0000

1.0000

1.0000

1.0000

S =

1

%Exercise 3

type solvesystem

function [] = solvesystem(A,b)

% This function returns the solution of a linear system using three methods

% The result are all subtracted from one another and placed into a new matrix

[~,n]=size(A);

format long

if (rank(A) < rank([A b]))

disp('The system is inconsistent')

else if (rank(A) < n)

disp('The system is consistent, but the solution is not unique')

else

disp('The system has a unique solution')

x1 = A\b;

x2 = inv(A)\*b;

x3\_sub = rref([A b]);

x3 = x3\_sub(:,end);

fprintf('The solutions obtained by different methods are the columns of ')

C = [x1,x2,x3]

n1 = norm(x1-x2);

n2 = norm(x2-x3);

n3 = norm(x3-x1);

fprintf('The norms of differences between solutions are the entries of ')

N = [n1;n2;n3]

end

end

end

A = magic(4);, b = randi(10,4,1);

solvesystem(A,b)

The system is inconsistent

A = magic(6);, b = ones(6,1);

solvesystem(A,b)

The system is consistent, but the solution is not unique

A = eye(5);, b = fix(10\*rand(5, 1));

solvesystem(A,b)

The system has a unique solution

The solutions obtained by different methods are the columns of

C =

9 9 9

4 4 4

8 8 8

1 1 1

4 4 4

The norms of differences between solutions are the entries of

N =

0

0

0

A= randi(10,4,4);, b = transpose([1:4]);

solvesystem(A,b)

The system has a unique solution

The solutions obtained by different methods are the columns of

C =

-1.443750000000000 -1.443750000000001 -1.443750000000000

0.378125000000000 0.378125000000000 0.378125000000000

0.956250000000000 0.956250000000000 0.956250000000000

-0.187500000000000 -0.187500000000000 -0.187500000000000

The norms of differences between solutions are the entries of

N =

1.0e-15 \*

0.289776716758409

0.461110253475620

0.343317509889168

A = magic(3);, b = randi(10,3,1);

solvesystem(A,b)

The system has a unique solution

The solutions obtained by different methods are the columns of

C =

0.577777777777778 0.577777777777778 0.577777777777778

0.911111111111111 0.911111111111111 0.911111111111111

-0.755555555555556 -0.755555555555555 -0.755555555555556

The norms of differences between solutions are the entries of

N =

1.0e-15 \*

0.157009245868378

0.157009245868378

0

format, A = hilb(6);, b = ones(6,1);

solvesystem(A,b)

The system has a unique solution

The solutions obtained by different methods are the columns of

C =

1.0e+03 \*

-0.006000000000904 -0.006000000000858 -0.006000000000000

0.210000000027077 0.210000000025611 0.210000000000000

-1.680000000188502 -1.680000000178348 -1.680000000000000

5.040000000499741 5.040000000472879 5.040000000000000

-6.300000000559204 -6.300000000528758 -6.300000000000000

2.772000000222627 2.772000000210363 2.772000000000000

The norms of differences between solutions are the entries of

N =

1.0e-06 \*

0.043636350241513

0.761522935201460

0.805157929418675

% For part(c) you a left multiplying vector b by an identity matrix of

% the correct size. This results in a solution to the linear system, which

% is equivelent to vector b. This is why the norm matrix entries are

% all zeros.

% The larger the hilbert matrix is the samller the determinate of the matrix

% is. The closer a determinate is to zero the less likely it is to be

% inveritble. The System is getting closer and closer to not have a unique

% solution, which is causing the methods to produce drastically different

% answers.

%Exercise 4

type homobasis\_b

function [C, p] = homobasis\_b(A,b)

m = size(A,1);

n = size(A,2);

red\_ech\_form = rref(A);

[~, pivot\_c] = rref(A);

S = 1:n;

P = setdiff(S,pivot\_c); %nonpivot columns

q = length(P);

p\_rows = any(red\_ech\_form,2);

B = -red\_ech\_form(p\_rows,P);

if q == 0

fprintf('the homogeneous system has only the trivial solution\n');

C = zeros(n,1);

else

disp('the homogenenous system has non-trivial solutions')

for i=P

fprintf('A free variable is x%d\n',i)

end

C(pivot\_c,:) = B;

C(P,:) = eye(q);

logic = A\*C < (10^(-7));

if size(C,1) == n && size(C,2) == q && rank(C) == q && all(logic(:))

fprintf('C is a basis for the solution set of the corresponding homogenous system\n');

R = rref([A b]);

p(pivot\_c) = R(1:n-q,n+1);

p(P) = 0;

p = transpose(p);

disp('The solution set of Ax=b is Span of the columns of C translated by vector p, where')

else

fprintf('Not a basis? Impossible!\n');

end

end

type nonhomogen

function x = nonhomogen(A,b)

[~,n]=size(A);

fprintf('Reduced echelon form of [A b] is ')

R=rref([A,b])

if rank([A b])==rank(A) && n == rank(A)

disp('The system has a unique solution')

x=A\b;

elseif rank([A b])==rank(A) && ~all(any(R,2))

disp('There are infinitely many solutions')

[C,p]=homobasis\_b(A,b)

syms Col(C), syms p

x=Col(C)+p;

else

disp('The system is inconsistent')

x=[];

end

end

%a

A=[1 2 -3], b=randi(10,1,1)

A =

1 2 -3

b =

7

x = nonhomogen(A,b)

Reduced echelon form of [A b] is R =

1 2 -3 7

The system is inconsistent

x =

[]

%b

A=magic(3), b=randi(10,3,1)

A =

8 1 6

3 5 7

4 9 2

b =

1

9

10

x = nonhomogen(A,b)

Reduced echelon form of [A b] is R =

1.0000 0 0 -0.5139

0 1.0000 0 1.1944

0 0 1.0000 0.6528

The system has a unique solution

x =

-0.5139

1.1944

0.6528

%c

A=magic(4), b=randi(10,4,1)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

b =

7

8

8

4

x = nonhomogen(A,b)

Reduced echelon form of [A b] is R =

1 0 0 1 0

0 1 0 3 0

0 0 1 -3 0

0 0 0 0 1

The system is inconsistent

x =

[]

%d

B=[0 1 2 3;0 2 4 6]; A=[B; eye(4)], b=sum(A,2)

A =

0 1 2 3

0 2 4 6

1 0 0 0

0 1 0 0

0 0 1 0

0 0 0 1

b =

6

12

1

1

1

1

x = nonhomogen(A,b)

Reduced echelon form of [A b] is R =

1 0 0 0 1

0 1 0 0 1

0 0 1 0 1

0 0 0 1 1

0 0 0 0 0

0 0 0 0 0

The system has a unique solution

x =

1.0000

1.0000

1.0000

1.0000

%e

A=[0 1 0 2 0 3; 0 2 0 4 0 6; 0 4 0 8 0 6], b=ones(3,1)

A =

0 1 0 2 0 3

0 2 0 4 0 6

0 4 0 8 0 6

b =

1

1

1

x = nonhomogen(A,b)

Reduced echelon form of [A b] is R =

0 1 0 2 0 0 0

0 0 0 0 0 1 0

0 0 0 0 0 0 1

The system is inconsistent

x =

[]

%f

A=[0 1 0 2 0 3; 0 2 0 4 0 6; 0 4 0 8 0 6], b=sum(A,2)

A =

0 1 0 2 0 3

0 2 0 4 0 6

0 4 0 8 0 6

b =

6

12

18

x = nonhomogen(A,b)

Reduced echelon form of [A b] is R =

0 1 0 2 0 0 3

0 0 0 0 0 1 1

0 0 0 0 0 0 0

There are infinitely many solutions

the homogenenous system has non-trivial solutions

A free variable is x1

A free variable is x3

A free variable is x4

A free variable is x5

C is a basis for the solution set of the corresponding homogenous system

The solution set of Ax=b is Span of the columns of C translated by vector p, where

C =

1 0 0 0

0 0 -2 0

0 1 0 0

0 0 1 0

0 0 0 1

0 0 0 0

p =

0

3

0

0

0

1

x =

p + Col(C)

%g

A=[0 0 1 2 3;0 0 2 4 5], b=A(:,end)

A =

0 0 1 2 3

0 0 2 4 5

b =

3

5

x = nonhomogen(A,b)

Reduced echelon form of [A b] is R =

0 0 1 2 0 0

0 0 0 0 1 1

The system is inconsistent

x =

[]

%h

A=[0 0 1 2 3;0 0 2 4 6], b=A(:,end)

A =

0 0 1 2 3

0 0 2 4 6

b =

3

6

x = nonhomogen(A,b)

Reduced echelon form of [A b] is R =

0 0 1 2 3 3

0 0 0 0 0 0

There are infinitely many solutions

the homogenenous system has non-trivial solutions

A free variable is x1

A free variable is x2

A free variable is x4

A free variable is x5

C is a basis for the solution set of the corresponding homogenous system

The solution set of Ax=b is Span of the columns of C translated by vector p, where

C =

1 0 0 0

0 1 0 0

0 0 -2 -3

0 0 1 0

0 0 0 1

p =

0

0

3

0

0

x =

p + Col(C)

%Exercise 5

type areavol

function D=areavol(A)

[m,n]=size(A);

if rank(A) == size(A,2)

if n==2

D = abs(det(A));

disp('The area of the parallelogram is')

else

D = abs(det(A));

disp('The volume of the parallelpiped is')

end

else

if n==2

disp('The parallelogram cannot be built')

D=0;

else

disp('The parallelepiped cannot be built')

D=0;

return

end

end

end

%(a)

A=randi(10,2,2)

A =

7 3

1 6

D=areavol(A)

The area of the parallelogram is

D =

39

%(b)

A=fix(10\*rand(3,3))

A =

9 9 8

9 9 1

1 4 4

D=areavol(A)

The volume of the parallelpiped is

D =

189

%(c)

A=magic(3)

A =

8 1 6

3 5 7

4 9 2

D=areavol(A)

The volume of the parallelpiped is

D =

360

%(d)

B=randi([-10,10], 2, 1); A=[B,2\*B]

A =

9 18

6 12

D=areavol(A)

The parallelogram cannot be built

D =

0

%(e)

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

A =

10 7 17

3 9 12

-10 4 -6

D=areavol(A)

The parallelepiped cannot be built

D =

0

% Exercise 6

type vrsh

function VS = vrsh(k)

% VS returns a vertical shear

e = eye(2);

e(2,1) = k;

VS = e;

end

type hrsh

function HS = hrsh(k)

% HS returns a horizontal shear

e = eye(2);

e(1,2) = k;

HS = e;

end

RX=[1 0; 0 -1]

RX =

1 0

0 -1

RY=[-1 0; 0 1]

RY =

-1 0

0 1

RO=[-1 0; 0 -1]

RO =

-1 0

0 -1

RS=[0 1; 1 0]

RS =

0 1

1 0

RA=[0 -1; -1 0]

RA =

0 -1

-1 0

type transf

function C = transf(A,E)

E=A\*E; %Multiplies the matrix A by matrix E

x=E(1,:); %Sets x equal to the first row of matrix E

y=E(2,:); %Sets y equal to the second row of matrix E

plot(x,y) %Plots vector y versus vector x

v=[-5 5 -5 5]; %Creates a matrix v that has a row of values -5, 5, -5, and 5.

axis(v) %Sets a scale for the x and y axis with -5 being the min and 5 being the max

grid %Creates a grid for the current axes

C=E; %Sets C equal to the matrix E

grid %Creates a grid for the current axes

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)

A =

1 0

0 1

hold

[\_Warning: MATLAB has disabled some advanced graphics rendering

features by switching to software OpenGL. For more

information, click <a href="matlab:opengl('problems')">here</a>.]\_

Current plot held

grid

C = transf(A,E)

C =

0 1 1 0 0

0 0 1 1 0

E=C;

k=2;

HS= hrsh(k)

HS =

1 2

0 1

A=HS;

C=transf(A,E)

C =

0 1 3 2 0

0 0 1 1 0

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)

A =

1 0

0 1

hold

Current plot held

grid

C=transf(A,E)

C =

0 1 1 0 0

0 0 1 1 0

E=C;

k=4;

VS = vrsh(k)

VS =

1 0

4 1

A=VS;

C=transf(A,E)

C =

0 1 1 0 0

0 4 5 1 0

C=transf(RX,C)

C =

0 1 1 0 0

0 -4 -5 -1 0

C=transf(RY,C)

C =

0 -1 -1 0 0

0 -4 -5 -1 0

C=transf(RX,C)

C =

0 -1 -1 0 0

0 4 5 1 0

C=transf(RS,C)

C =

0 4 5 1 0

0 -1 -1 0 0

C=transf(RX,C)

C =

0 4 5 1 0

0 1 1 0 0

C=transf(RY,C)

C =

0 -4 -5 -1 0

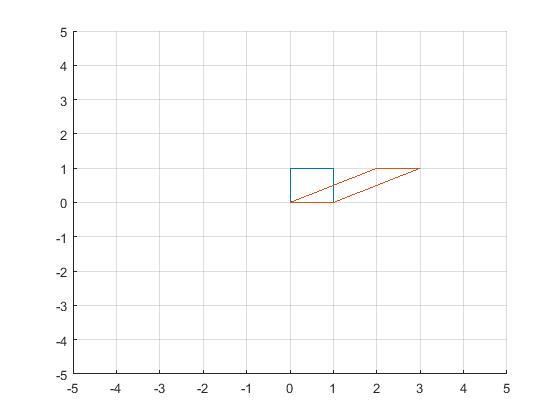
0 1 1 0 0

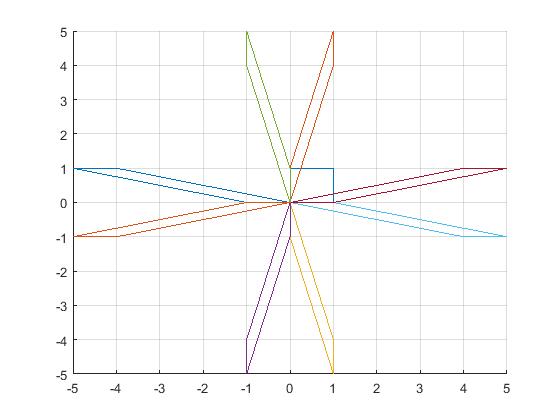
C=transf(RX,C)

C =

0 -4 -5 -1 0

0 -1 -1 0 0





%Exercise 7

type cofactor

function C = cofactor( a )

[n, m] = size(a);

for i = 1:n

for j = 1:n

aij = a(~ismember(1:n,i),~ismember(1:n,j));

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

end

end

end

type determine

function D = determine( a, C )

[n,m] = size(a);

aC = a .\* C;

epsilon = 10^(-7);

D1 = sum(aC, 1);

D2 = sum(aC, 2);

if(all(abs(D1 - transpose(D2)) <= epsilon))

D = D1(1);

else

D = [];

disp("Oh no! There must be a problem with my code!");

end

end

type inverse

function B = inverse(a, C, D)

%Rank is used to determine invertibility because

%the invertible matrix theorem states that if rank(a) == n if and only if the matrix is invertible.

%we choose not to determine invertibility by checking if det(a) == 0 because of floating point errors that may occur.

[n, m] = size(a);

if rank(a) == n

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

else

disp("empty\_matrix");

B = [];

end

end

%(a)

a = diag([1,2,3,4,5])

a =

1 0 0 0 0

0 2 0 0 0

0 0 3 0 0

0 0 0 4 0

0 0 0 0 5

C = cofactor(a)

C =

120 0 0 0 0

0 60 0 0 0

0 0 40 0 0

0 0 0 30 0

0 0 0 0 24

D = determine(a,C)

D =

120

det(a)

ans =

120

B = inverse(a,C,D)

B =

1 0 0 0 0

0 1/2 0 0 0

0 0 1/3 0 0

0 0 0 1/4 0

0 0 0 0 1/5

inv(a)

ans =

1 0 0 0 0

0 1/2 0 0 0

0 0 1/3 0 0

0 0 0 1/4 0

0 0 0 0 1/5

%(b)

a = ones(4)

a =

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

D = determine(a,C)

D =

0

det(a)

ans =

0

B = inverse(a,C,D)

empty\_matrix

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 signifies that matrix is not invertible

%(c)

a = magic(3)

a =

8 1 6

3 5 7

4 9 2

C = cofactor(a)

C =

-53 22 7

52 -8 -68

-23 -38 37

D = determine(a,C)

D =

-360

det(a)

ans =

-360

B = inverse(a,C,D)

B =

53/360 -13/90 23/360

-11/180 1/45 19/180

-7/360 17/90 -37/360

inv(a)

ans =

53/360 -13/90 23/360

-11/180 1/45 19/180

-7/360 17/90 -37/360

%(d)

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 =

1/628292358729

det(a)

ans =

-1/689889648801

% Difference is due to both being effectively zero.

B = inverse(a,C,D)

empty\_matrix

B =

[]

inv(a)

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

ans =

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

% \*'s as value indicates that matrix is not invertible

%(e)

format, format compact

a = hilb(5)

a =

1.0000 0.5000 0.3333 0.2500 0.2000

0.5000 0.3333 0.2500 0.2000 0.1667

0.3333 0.2500 0.2000 0.1667 0.1429

0.2500 0.2000 0.1667 0.1429 0.1250

0.2000 0.1667 0.1429 0.1250 0.1111

C = cofactor(a)

C =

1.0e-06 \*

0.0001 -0.0011 0.0039 -0.0052 0.0024

-0.0011 0.0180 -0.0709 0.1008 -0.0472

0.0039 -0.0709 0.2976 -0.4409 0.2126

-0.0052 0.1008 -0.4409 0.6719 -0.3307

0.0024 -0.0472 0.2126 -0.3307 0.1653

D = determine(a,C)

D =

3.7493e-12

det(a)

ans =

3.7493e-12

B = inverse(a,C,D)

B =

1.0e+05 \*

0.0002 -0.0030 0.0105 -0.0140 0.0063

-0.0030 0.0480 -0.1890 0.2688 -0.1260

0.0105 -0.1890 0.7938 -1.1760 0.5670

-0.0140 0.2688 -1.1760 1.7920 -0.8820

0.0063 -0.1260 0.5670 -0.8820 0.4410

inv(a)

ans =

1.0e+05 \*

0.0002 -0.0030 0.0105 -0.0140 0.0063

-0.0030 0.0480 -0.1890 0.2688 -0.1260

0.0105 -0.1890 0.7938 -1.1760 0.5670

-0.0140 0.2688 -1.1760 1.7920 -0.8820

0.0063 -0.1260 0.5670 -0.8820 0.4410

diary off