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 # \_\_\_\_\_13\_\_\_\_\_\_

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

\_1.\_\_\_\_\_\_Koby Miller\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_2.\_\_\_\_\_\_Sean O’Reilly\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ­­­

\_3.\_\_\_\_\_\_Nicholas Miller\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_4.\_\_\_\_\_\_Carl Simpron\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_5.\_\_\_\_\_\_Daniel Silva \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_6.\_\_\_\_\_\_\_Keval Patel\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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

%Part 1

%Exercise 1

type ele1

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

% This creates the first elementary matrix that is used to add a multiple

% of one row to another row

X = eye(n);

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

E1=X;

end

type ele2

function E2=ele2(n,i,j)

% This creates the second elementary matrix that is used to swap rows of

% a matrix

X = eye(n);

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

E2=X;

end

type ele3

function E3=ele3(n,j,k)

% This creates the third elemetary matrix that multiplies one whole row

% by a constant

X=eye(n);

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

E3=X;

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;

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

E3 = ele3(3,3,-1/2)

E3 =

1 0 0

0 1 0

0 0 -1/2

A5 = E3\*A4

A5 =

1 0 2

0 1 -1

0 0 1

%Part 2

%Exercise#2

type inverses

function D = inverses(A)

[n,m] = size(A);

if(rank(A) == n)

D = rref([A eye(n)]);

D = D(:, n+1:2\*n);

else

D = [];

disp('Matrix A is not invertable')

end

end

%(a)

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

D = inverses(A)

D =

-19 -14 0 7

-549 -401 -2 196

267 195 1 -95

-278 -203 -1 99

%(b)

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

D = inverses(A)

Matrix A is not invertable

D =

[]

%(c)

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

D = inverses(A)

D =

-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

%(d)

A = magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

D = inverses(A)

Matrix A is not invertable

D =

[]

%Testing part a

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

D = inv(A)

D =

-19.0000 -14.0000 -0.0000 7.0000

-549.0000 -401.0000 -2.0000 196.0000

267.0000 195.0000 1.0000 -95.0000

-278.0000 -203.0000 -1.0000 99.0000

%Matches output

%Testing part b

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

D = inv(A)

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

D =

Inf Inf Inf Inf

Inf Inf Inf Inf

Inf Inf Inf Inf

Inf Inf Inf Inf

%Matches output (inverse does not exist)

%Testing part c

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

D = inv(A)

D =

-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

%Matches output

%Testing part d

A = magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

D = inv(A)

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

1.306145e-17.]\_

D =

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

%This is inaccurate because the determinant calculated by matlabs function was not zero

%It calculated det(A) as 1.0e-14 which is very close, but not zero, so it gave an output even though the inverse does not exist.

%Part 3

%Exercise3

function [C,N] = solvesys(A)

format long

if det(A) == 0,

C=[];

N=[];

S = 'The system is either inconsistent or the solution is not unique.';

disp(S);

else,

C=[];

N=[];

n = size(A);

n = n(1,1);

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

A1 = A\b;

C(:,1) = A1(:,1);

A2 = inv(A)\*b;

C(:,2) = A2(:,1);

A3 = rref([A b]);

C(:,3) = A3(:,4);

N(1,1) = norm(A1 - A2);

N(1,2) = norm(A2 - A3);

N(1,3) = norm(A3 - A1);

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 =

9

1

2

1

1

8

[\_Warning: Matrix is close to singular or badly

scaled. Results may be inaccurate. RCOND =

5.601125e-18.]\_

[\_> In <a href="matlab:matlab.internal.language.introspective.errorDocCallback('solvesys', 'M:\solvesys.m', 16)" style="font-weight:bold">solvesys</a> (<a href="matlab: opentoline('M:\solvesys.m',16,0)">line 16</a>)

]\_

[\_Warning: Matrix is close to singular or badly

scaled. Results may be inaccurate. RCOND =

5.601125e-18.]\_

[\_> In <a href="matlab:matlab.internal.language.introspective.errorDocCallback('solvesys', 'M:\solvesys.m', 19)" style="font-weight:bold">solvesys</a> (<a href="matlab: opentoline('M:\solvesys.m',19,0)">line 19</a>)]\_

C =

1.0e+15 \*

Columns 1 through 2

-4.503599627370496 -4.503599627370496

-4.503599627370496 -4.503599627370497

2.251799813685248 2.251799813685248

4.503599627370496 4.503599627370498

4.503599627370496 4.503599627370495

-2.251799813685248 -2.251799813685247

Column 3

0

0

0

0.000000000000001

0

0

N =

1.0e+16 \*

Columns 1 through 2

0.000000000000000 2.527639021923478

Column 3

2.527639021923477

%(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 =

5

5

1

8

6

C =

5 5 0

5 5 0

1 1 0

8 8 1

6 6 0

N =

Columns 1 through 2

0 26.574129317027527

Column 3

26.574129317027527

%(c)

A = randi(20,4,4)

A =

8 5 9 10

11 3 1 10

9 4 19 7

2 5 19 19

[C,N] = solvesys(A)

b =

3

1

7

3

C =

Columns 1 through 2

0.230704272926871 0.230704272926871

0.059111636547880 0.059111636547880

0.321736193210606 0.321736193210606

-0.203681810504982 -0.203681810504982

Column 3

0

0

0

1.000000000000000

N =

Columns 1 through 2

0.000000000000000 1.340113438338961

Column 3

1.340113438338961

%(d)

A = magic(3)

A =

8 1 6

3 5 7

4 9 2

[C,N] = solvesys(A)

b =

2

4

0

C =

Columns 1 through 2

-0.283333333333333 -0.283333333333333

-0.033333333333333 -0.033333333333333

0.716666666666667 0.716666666666667

Column 3

-0.283333333333333

-0.033333333333333

0.716666666666667

N =

Columns 1 through 2

0.000000000000000 1.690883088068309

Column 3

1.690883088068309

%(e)

format rat

A = hilb(6)

A =

Columns 1 through 3

1 1/2 1/3

1/2 1/3 1/4

1/3 1/4 1/5

1/4 1/5 1/6

1/5 1/6 1/7

1/6 1/7 1/8

Columns 4 through 6

1/4 1/5 1/6

1/5 1/6 1/7

1/6 1/7 1/8

1/7 1/8 1/9

1/8 1/9 1/10

1/9 1/10 1/11

[C,N] = solvesys(A)

b =

1

9

9

5

0

2

C =

1.0e+07 \*

Columns 1 through 2

-0.001873800000157 -0.001873800000150

0.056259000004737 0.056259000004509

-0.393036000033107 -0.393036000031431

1.044540000087997 1.044540000083415

-1.170666000098647 -1.170666000093420

0.466527600039326 0.466527600037219

Column 3

0

0

0

0.000000100000000

0

0

N =

1.0e+07 \*

Columns 1 through 2

0.000000000007458 4.125639281819805

Column 3

4.125639281838071

%interestingly, the first two cols of part b's n are equal to vector b

%part e has different norms because the deteminant of hilb(6) is a tiny fraction

%Part 4

% Exercise #4

type arevol

function [D,A] = arevol(B)

[m,n] = size(B);

for i=1:m

for j=2:n

A(i,j - 1) = B(i,j) - B(i,1);

end

end

[m,n] = size(A);

D = abs(det(A));

D = closetozeroroundoff(D);

if (m == 2 && n == 2 && D == 0)

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

return;

else if (m == 3 && n == 3 && D == 0)

disp('The points lie on the same line and no parallelepiped can be built');

return;

end

end

if (m == 2 && n == 2 && D ~= 0)

disp('The area of the parallelogram is ');

disp(D);

else

disp('The volume of the parallelepiped is ');

disp(D);

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;

end

% (a)

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

B =

-5 10 -7

1 10 10

[D,A] = arevol(B)

A =

15 -2

9 9

The area of the parallelogram is

153

% (b)

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

B =

10 -8 6 -10

0 -2 10 7

6 9 3 9

[D,A] = arevol(B)

A =

-18 -4 -20

-2 10 7

3 -3 3

The volume of the parallelepiped is

546.0000

% (c)

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

X =

4

5

B =

4 -4 8

5 -5 10

[D,A] = arevol(B)

A =

-8 4

-10 5

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

% (d)

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

X =

5

-2

3

Y =

-7

4

-10

B =

5 -7 -2 12

-2 4 2 -6

3 -10 -7 13

[D,A] = arevol(B)

A =

-12 -7 7

6 4 -4

-13 -10 10

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

%part 5

%Exercise 5

format compact

R1 = [-1,0;0,1]

R1 =

-1 0

0 1

R2 = [1,0;0,-1]

R2 =

1 0

0 -1

VS = [1,0;2,1]

VS =

1 0

2 1

type transf

function C = transf(A,E)

%multiply the two matices

E=A\*E;

%storing the value of row 1 of E to x and row2 or E to Y

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

%plot the matrix

plot(x,y)

%creating a vector

v=[-5 5 -5 5];

%sets the limits and ratios of the vector

axis(v)

grid

C=E;

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;

C = transf(A,E)

C =

0 1 1 0 0

0 2 3 1 0

E = C;

A = R1;

C = transf(A,E)

A = R1;

C = transf(A,E)

C =

0 -1 -1 0 0

0 2 3 1 0

A = R2;

E = C;

C = transf(A,E)

C =

0 -1 -1 0 0

0 -2 -3 -1 0

A = R1;

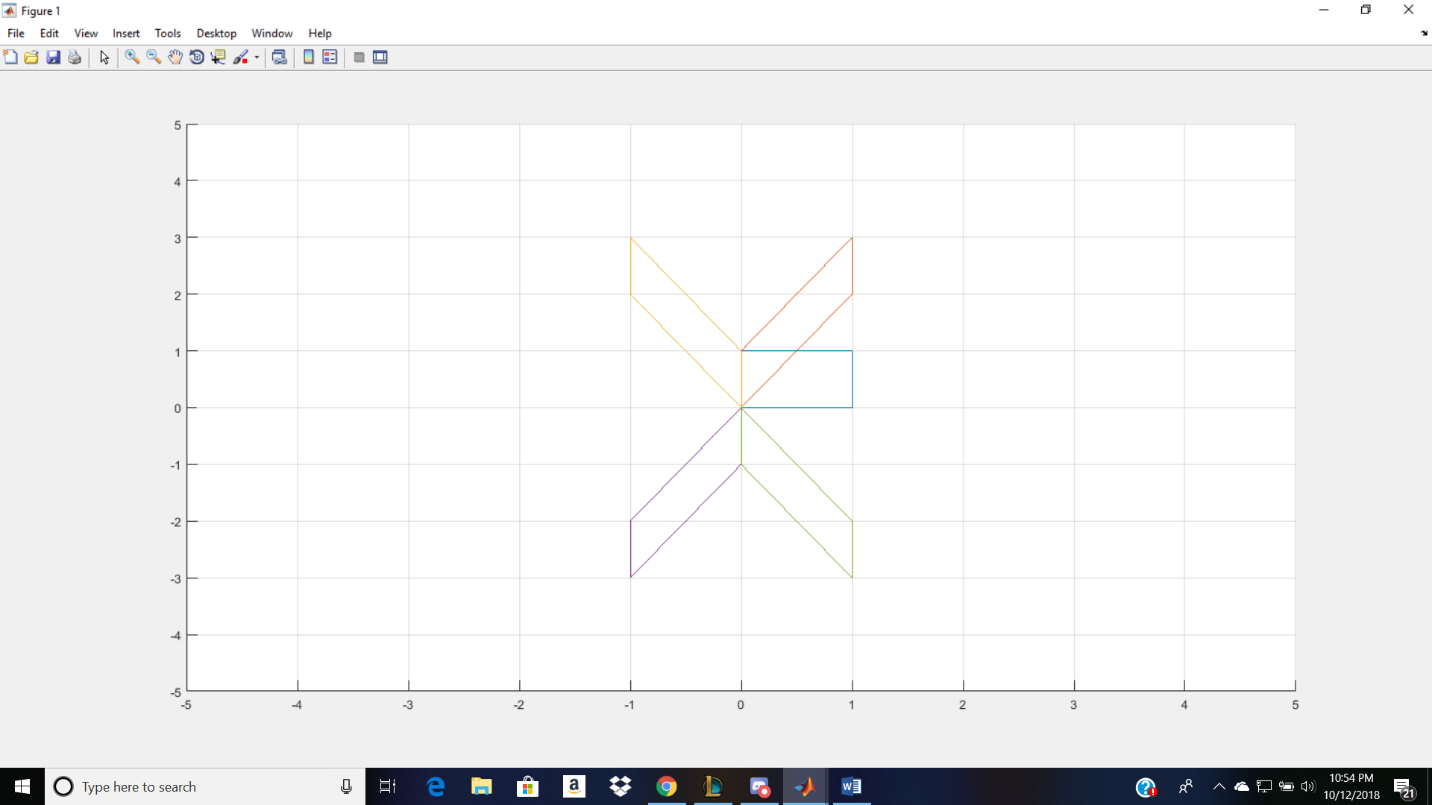
E = C;

C = transf(A,E)

C =

0 1 1 0 0

0 -2 -3 -1 0



%Exercise 6

type cofactor

function [C] = cofactor(a)

%COFACTOR

C = [];

for i =1:1:size(a, 1)

for j = 1:1:size(a,2)

A = a;

A(i,:) = [];

A(:,j) = [];

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

end

end

end

type determine

function [D] = determine(a, C)

%DETERMINE Summary of this function goes here

% Detailed explanation goes here

D1 = [];

D2 = [];

for i =1:1:size(a, 1)

j= 1;

D1(i, 1) = a(i,j)\*C(i,j);

end

for j =1:1:size(a,2)

i = 1;

D2(j,1) = a(i,j)\*C(i,j);

end

if D1 ~= D2

fprintf('There is a problem with my code!')

else

D = sum(D1);

end

end

type inverse

function [B] = inverse(a, C, D)

%INVERSE Summary of this function goes here

% Detailed explanation goes here

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

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

else

B = [];

end

diary on

format compact

format rat

%a)

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

a =

Columns 1 through 3

1 0 0

0 2 0

0 0 3

0 0 0

Column 4

0

0

0

4

C=cofactor(a)

C =

Columns 1 through 3

24 0 0

0 12 0

0 0 8

0 0 0

Column 4

0

0

0

6

D=determine(a,C)

D =

24

det(a)

ans =

24

%Determinants are the same

B = inverse(a,C,D)

B =

Columns 1 through 3

1 0 0

0 1/2 0

0 0 1/3

0 0 0

Column 4

0

0

0

1/4

inv(a)

ans =

Columns 1 through 3

1 0 0

0 1/2 0

0 0 1/3

0 0 0

Column 4

0

0

0

1/4

%Inverses are the same

%b)

a = ones(5)

a =

Columns 1 through 3

1 1 1

1 1 1

1 1 1

1 1 1

1 1 1

Columns 4 through 5

1 1

1 1

1 1

1 1

1 1

C=cofactor(a)

C =

Columns 1 through 3

0 0 0

0 0 0

0 0 0

0 0 0

0 0 0

Columns 4 through 5

0 0

0 0

0 0

0 0

0 0

D=determine(a,C)

D =

0

det(a)

ans =

0

%Determinants are the same

B=inverse(a,C,D)

B =

[]

inv(a)

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

ans =

Columns 1 through 3

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

Columns 4 through 5

1/0 1/0

1/0 1/0

1/0 1/0

1/0 1/0

1/0 1/0

%The answers are different but mean the same thing. Essentially the inv program divides the transposed matrix by 0 which obviously does not exist.

%c)

a = magic(5)

a =

Columns 1 through 3

17 24 1

23 5 7

4 6 13

10 12 19

11 18 25

Columns 4 through 5

8 15

14 16

20 22

21 3

2 9

C=cofactor(a)

C =

Columns 1 through 3

-25025 218725 -153400

259350 -189150 15600

-179400 -23400 15600

5850 64350 15600

17225 7475 184600

Columns 4 through 5

23725 13975

-33150 25350

54600 210600

220350 -228150

-187525 56225

D=determine(a,C)

D =

5070000

det(a)

ans =

5070000

%Determinants are the same

B=inverse(a,C,D)

B =

Columns 1 through 3

-77/15600 133/2600 -23/650

89/2063 -97/2600 -3/650

-59/1950 1/325 1/325

73/15600 -17/2600 7/650

43/15600 1/200 27/650

Columns 4 through 5

3/2600 53/15600

33/2600 23/15600

1/325 71/1950

113/2600 -577/15600

-9/200 98/8837

inv(a)

ans =

Columns 1 through 3

-77/15600 133/2600 -23/650

89/2063 -97/2600 -3/650

-59/1950 1/325 1/325

73/15600 -17/2600 7/650

43/15600 1/200 27/650

Columns 4 through 5

3/2600 53/15600

33/2600 23/15600

1/325 71/1950

113/2600 -577/15600

-9/200 98/8837

%Inverses are the same

%d)

a=magic(4)

a =

Columns 1 through 3

16 2 3

5 11 10

9 7 6

4 14 15

Column 4

13

8

12

1

C=cofactor(a)

C =

Columns 1 through 3

-136 -408 408

-408 -1224 1224

408 1224 -1224

136 408 -408

Column 4

136

408

-408

-136

D=determine(a,C)

D =

1/586406201481

det(a)

ans =

-1/689889648801

%Both values are essentially the same, 0. What this means is that matlab calculates the determinant differently than we did.

B=inverse(a,C,D)

B =

[]

inv(a)

[\_Warning: Matrix is close to singular or badly scaled.

Results may be inaccurate. RCOND = 1.306145e-17.]\_

ans =

Columns 1 through 3

\* \* \*

\* \* \*

\* \* \*

\* \* \*

Column 4

\*

\*

\*

\*

%No inverse exists because the determinant is 0. The asterisk indicate a very small number, which we can take to be 0. This essentially means that no inverse exists.

%e)

a =hilb(4)

a =

Columns 1 through 3

1 1/2 1/3

1/2 1/3 1/4

1/3 1/4 1/5

1/4 1/5 1/6

Column 4

1/4

1/5

1/6

1/7

C=cofactor(a)

C =

Columns 1 through 3

1/378000 -1/50400 1/25200

-1/50400 1/5040 -1/2240

1/25200 -1/2240 3/2800

-1/43200 1/3600 -1/1440

Column 4

-1/43200

1/3600

-1/1440

1/2160

D=determine(a,C)

D =

1/6048000

det(a)

ans =

1/6048000

%Determinants match

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

%inverses are the same.

diary off