Koby Miller

Project 0

diary on

format compact

A = [-2 -1 8 3; 0 -2 3 1; -3 -7 5 4]

A =

-2 -1 8 3

0 -2 3 1

-3 -7 5 4

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

B =

1 -2

3 -1

1 0

X = [-2;3;5]

X =

-2

3

5

x = [1 2 3 4]

x =

1 2 3 4

y = [1;4;6;3]

y =

1

4

6

3

A

A =

-2 -1 8 3

0 -2 3 1

-3 -7 5 4

save Project0

size(A)

ans =

3 4

size(B)

ans =

3 2

size(X)

ans =

3 1

size(x)

ans =

1 4

size(y)

ans =

4 1

%The matrices x and y are different sizes

%This is because x has 4 columns and only 1 row

%But y has 1 column and 4 rows

F = [5 2 -3;4 3 -2;0 -1 6;1 0 -2]

F =

5 2 -3

4 3 -2

0 -1 6

1 0 -2

A(1,3)

ans =

8

%this gave the answer to the number in the first row, and third column of matrix A

A(:,3)

ans =

8

3

5

%this gave the whole third column of matrix A

A(2,:)

ans =

0 -2 3 1

%This gave the whole second row of matrix A

A([1 2], [3 4])

ans =

8 3

3 1

%In matrix A, it gave the entires in row 1 and 2 that were also in columns 3 and 4

F(:,4)=[-1 1 -4 3]

F =

5 2 -3 -1

4 3 -2 1

0 -1 6 -4

1 0 -2 3

%This added another column to matrix F

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

F =

5 1 -3 -3

4 3 -2 1

0 2 6 -4

1 0 -2 3

%In the first row, the numbers in column 2 and 4 were replaced with 1 and -3

%Also in the third row, the numbers in column 2 and 4 were replaced by 2 and -4

F([2 3], :) = A([1 3], :)

F =

5 1 -3 -3

-2 -1 8 3

-3 -7 5 4

1 0 -2 3

%This replaced the second and third row of F with the first and third row of A

F(:, [1 2])=F(:, [2 1])

F =

1 5 -3 -3

-1 -2 8 3

-7 -3 5 4

0 1 -2 3

%this swapped column 1 and 2 of matrix F

F(:,[1])=y(:,[1])

F =

1 5 -3 -3

4 -2 8 3

6 -3 5 4

3 1 -2 3

%this swithces the first column of F with the vector y

F([1 2],:) = F([2 1],:)

F =

4 -2 8 3

1 5 -3 -3

6 -3 5 4

3 1 -2 3

%this swiches the first and second row of matrix F

save Project0

[A B]

ans =

-2 -1 8 3 1 -2

0 -2 3 1 3 -1

-3 -7 5 4 1 0

%this added matrix B on to the right of matrix A

[B A]

ans =

1 -2 -2 -1 8 3

3 -1 0 -2 3 1

1 0 -3 -7 5 4

%This did the opposite in that it added A to the right side of B

[A x]

{Error using <a href="matlab:matlab.internal.language.introspective.errorDocCallback('horzcat')" style="font-weight:bold">horzcat</a>

Dimensions of arrays being concatenated are not

consistent.}

%The error explains itself, dimensions were not consistent

[A X]

ans =

-2 -1 8 3 -2

0 -2 3 1 3

-3 -7 5 4 5

%Added X to the right of A

[A ; y]

{Error using <a href="matlab:matlab.internal.language.introspective.errorDocCallback('vertcat')" style="font-weight:bold">vertcat</a>

Dimensions of arrays being concatenated are not

consistent.}

%Again, dimensions are not consistent

[A ; x]

ans =

-2 -1 8 3

0 -2 3 1

-3 -7 5 4

1 2 3 4

%This added matrix x to the bottm of A

eye(5)

ans =

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

%This created a 5x5 matrix with 1's on the diagonal and 0's everywhere else

zeros(3,4)

ans =

0 0 0 0

0 0 0 0

0 0 0 0

%this created a 3x4 matrix filled with 0's

ones(3,2)

ans =

1 1

1 1

1 1

%this created a 3x2 matrix filled with 1's

ones(5)

ans =

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

%this creates a 5x5 matrix filled wiht 1's

diag([1 2 5 6 7])

ans =

1 0 0 0 0

0 2 0 0 0

0 0 5 0 0

0 0 0 6 0

0 0 0 0 7

%This created a matrix that filled the diagonal line with the desiered numbers and filled the rest of the entries with 0's

diag([1 2 5 6 7],-1)

ans =

0 0 0 0 0 0

1 0 0 0 0 0

0 2 0 0 0 0

0 0 5 0 0 0

0 0 0 6 0 0

0 0 0 0 7 0

%This made another row of 0's above and to the right of the previous matrix

%It is not the same matrix, it just does the same thing except it creates the extra row and column

diag([1 2 5 6 7],2)

ans =

0 0 1 0 0 0 0

0 0 0 2 0 0 0

0 0 0 0 5 0 0

0 0 0 0 0 6 0

0 0 0 0 0 0 7

0 0 0 0 0 0 0

0 0 0 0 0 0 0

%This is similar to the last command but instead the columns were added to the left and bottom insted of the right an the top

A, diag(A), diag(diag(A))

A =

-2 -1 8 3

0 -2 3 1

-3 -7 5 4

ans =

-2

-2

5

ans =

-2 0 0

0 -2 0

0 0 5

%This first showed you Matrix A

%Then It gave you the diagnol of A

%Then it made a matrrix with the diagonal of A and 0's everywhere else

save Project0

magic(5)

ans =

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

%This creates a magic square

%A magic square is a matrix where each row, column and diagonal have the same sums

help magic

<strong>magic</strong> Magic square.

<strong>magic</strong>(N) is an N-by-N matrix constructed from the integers

1 through N^2 with equal row, column, and diagonal sums.

Produces valid magic squares for all N > 0 except N = 2.

<a href="matlab:doc magic">Reference page for magic</a>

hilb(5)

ans =

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

help hilb

<strong>hilb</strong> Hilbert matrix.

H = <strong>hilb</strong>(N) is the N-by-N matrix with elements 1/(i+j-1), which is a

famous example of a badly conditioned matrix. The INVHILB function

calculates the exact inverse.

H = <strong>hilb</strong>(N,CLASSNAME) returns a matrix of class CLASSNAME, which can be

either 'single' or 'double' (the default).

<strong>hilb</strong> is also a good example of efficient MATLAB programming

style, where conventional FOR or DO loops are replaced by

vectorized statements.

Example:

<strong>hilb</strong>(3) is

1.0000 0.5000 0.3333

0.5000 0.3333 0.2500

0.3333 0.2500 0.2000

See also <a href="matlab:help invhilb">invhilb</a>.

<a href="matlab:doc hilb">Reference page for hilb</a>

%\*\* I forgot to actually set the matrices below to the variables C,D and E

%\*\* When we use the matrices later I realize this a remake them there

eye(3)

ans =

1 0 0

0 1 0

0 0 1

diag([2 1 3],1)

ans =

0 2 0 0

0 0 1 0

0 0 0 3

0 0 0 0

ones(2,3)

ans =

1 1 1

1 1 1

V1=1:7

V1 =

1 2 3 4 5 6 7

V2=2:0.5:6.5

V2 =

Columns 1 through 5

2.0000 2.5000 3.0000 3.5000 4.0000

Columns 6 through 10

4.5000 5.0000 5.5000 6.0000 6.5000

V3=3:-1:-5

V3 =

3 2 1 0 -1 -2 -3 -4 -5

%These commands all just made one long row and

%you decided what number to start at, how much

% to increment by, and what number to count to

V4=-5:1

V4 =

-5 -4 -3 -2 -1 0 1

v5=10:-3:-2

v5 =

10 7 4 1 -2

V5=10:-3:-2

V5 =

10 7 4 1 -2

V6=5:-0.5:2

V6 =

Columns 1 through 5

5.0000 4.5000 4.0000 3.5000 3.0000

Columns 6 through 7

2.5000 2.0000

V7=0:0.4:4

V7 =

Columns 1 through 5

0 0.4000 0.8000 1.2000 1.6000

Columns 6 through 10

2.0000 2.4000 2.8000 3.2000 3.6000

Column 11

4.0000

C;

C

C =

1 0 0

0 1 0

0 0 1

%When you put the semicolon at the end of the command you do not see the results

%With no semicolon, you do see the results

R=434.1452

R =

434.1452

%Saved the number to R

format long, R

R =

4.341452000000000e+02

%It took R and displayed it to as many digits as it could

format short, R

R =

434.1452

%It took R and displayed it up to as many digits as it had

save Project0

format

A, A+A

A =

-2 -1 8 3

0 -2 3 1

-3 -7 5 4

ans =

-4 -2 16 6

0 -4 6 2

-6 -14 10 8

A, 2\*A

A =

-2 -1 8 3

0 -2 3 1

-3 -7 5 4

ans =

-4 -2 16 6

0 -4 6 2

-6 -14 10 8

A, B, A+B

A =

-2 -1 8 3

0 -2 3 1

-3 -7 5 4

B =

1 -2

3 -1

1 0

{Matrix dimensions must agree.}

%This explains why we have the error

B,E, B-2\*E

B =

1 -2

3 -1

1 0

{Undefined function or variable 'E'.}

%\*\*This is where I make C,D,and E

C= eye(3)

C =

1 0 0

0 1 0

0 0 1

D = diag([2 1 3],1)

D =

0 2 0 0

0 0 1 0

0 0 0 3

0 0 0 0

E = ones(2,3)

E =

1 1 1

1 1 1

%Now they are made so I can try again

B,E,B-2\*E

B =

1 -2

3 -1

1 0

E =

1 1 1

1 1 1

{Matrix dimensions must agree.}

%Again the error is because of the dimensions

x,X,x+X

x =

1 2 3 4

X =

-2

3

5

ans =

-1 0 1 2

4 5 6 7

6 7 8 9

x,y,x+y

x =

1 2 3 4

y =

1

4

6

3

ans =

2 3 4 5

5 6 7 8

7 8 9 10

4 5 6 7

%The reason some have error messages is because the dimensinos do not agree so they cannot preform the math

transpose(A)

ans =

-2 0 -3

-1 -2 -7

8 3 5

3 1 4

A\*D

ans =

0 -4 -1 24

0 0 -2 9

0 -6 -7 15

A.\*A

ans =

4 1 64 9

0 4 9 1

9 49 25 16

A.\*D

{Matrix dimensions must agree.}

A\*A

{Error using <a href="matlab:matlab.internal.language.introspective.errorDocCallback('mtimes')" style="font-weight:bold"> \* </a>

Incorrect dimensions for matrix multiplication. Check that

the number of columns in the first matrix matches the

number of rows in the second matrix. To perform

elementwise multiplication, use '.\*'.}

%The last two operations produce an error message

%The reason for the first is becasue the dimensions do not agree but they need to for .\*

%The reason for the second is becasue the dimensions do not match

G = [4 2 1; 3 1 6; 7 7 8]

G =

4 2 1

3 1 6

7 7 8

G^2

ans =

29 17 24

57 49 57

105 77 113

A^2

{Error using <a href="matlab:matlab.internal.language.introspective.errorDocCallback('mpower')" style="font-weight:bold"> ^ </a>

Incorrect dimensions for raising a matrix to a power.

Check that the matrix is square and the power is a scalar.

To perform elementwise matrix powers, use '.^'.}

G

G =

4 2 1

3 1 6

7 7 8

G\*G

ans =

29 17 24

57 49 57

105 77 113

%Doing G\*G is the equivalent to G^2

%the second command: A^2 produces an error becasue it is not a square matrix so the dimensions will not work to multiply

%the command you should use is .^

A.^2

ans =

4 1 64 9

0 4 9 1

9 49 25 16

rand(4)

ans =

0.8147 0.6324 0.9575 0.9572

0.9058 0.0975 0.9649 0.4854

0.1270 0.2785 0.1576 0.8003

0.9134 0.5469 0.9706 0.1419

rand(3,4)

ans =

0.4218 0.9595 0.8491 0.7577

0.9157 0.6557 0.9340 0.7431

0.7922 0.0357 0.6787 0.3922

randi(100, 2)

ans =

66 71

18 4

randi(10, 2, 4)

ans =

3 1 7 10

1 9 4 1

randi([10 40], 2, 4)

ans =

23 33 15 23

21 34 25 30

%The first matrix is a 4x4 and all the numbers are between 0 and 1 and are random

%The second matrix is a 3x4 matrix with all the entries are random numbers between 0 and 1

%The third matrixis a 2x2 matrix with random numbers between 0 and 100

%The fourth matrix is a 2x4 matrix with random numbers between 0 and 10

%The last matrix is a 2x4 with random numbers between 10 and 40

%One last note is that the last 3 matrices are intigers. They are full numbers with no decimal places

save Project0

5\*rand(3)

ans =

3.5468 3.3985 0.5950

3.7734 3.2755 2.4918

1.3801 0.8131 4.7987

-3+5\*rand(3)

ans =

-1.2981 0.7563 0.4954

-0.0737 -1.7245 1.4545

-1.8809 -0.4702 1.7965

% Between 4 and 10

4+6\*rand(2,3)

ans =

7.2833 4.8958 9.0443

4.8317 5.5450 5.5257

% Between 40 and 90

randi([40 90],2,3)

ans =

81 87 50

52 57 52

save Project0

% Exercise 3

type adds

function C=adds(A,B)

% This is the function which adds

% matrices A and B. It duplicates the MATLAB

% function A+B.

[m,n]=size(A);

[k,p]=size(B);

if m==k && n==p,

for i=1:m

for j=1:n

C(i,j)=A(i,j)+B(i,j);

end

end

else

disp('Error in using adds: matrices are not of the same size')

C=[];

end

end

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

A =

1 2 3

4 5 6

7 8 9

B=ones(2,3)

B =

1 1 1

1 1 1

C=adds(A,B)

Error in using adds: matrices are not of the same size

C =

[]

A=magic(3)

A =

8 1 6

3 5 7

4 9 2

B=ones(3)

B =

1 1 1

1 1 1

1 1 1

C=adds(A,B)

C =

9 2 7

4 6 8

5 10 3

save Project0

%Exercise 4

type sums

function A=sums(i,j)

% This is the function which creates a matrix

% where the number at each coordinate (i,j)

% is just i+j

if i<0 || j<0 %Checks if negative

disp('Error in using sums: both m and n must be positive integer. ')

A=[];

return

end

if mod(i,1)~= 0 || mod(j,1) ~=0 %Checks if whole number

disp('Error in using sums: both m and n must be positive integer. ')

A=[];

return

end

B = zeros(i,j);

for a=1:i

for b=1:j

B(a,b)=a+b;

end

end

A=B;

end

%(a)

m=3.5,n=4

m =

3.5000

n =

4

A=sums(m,n)

Error in using sums: both m and n must be positive integer.

A =

[]

%(b)

m=5,n=2.6

m =

5

n =

2.6000

A=sums(m,n)

Error in using sums: both m and n must be positive integer.

A =

[]

%(c)

m=3,n=5

m =

3

n =

5

A=sums(m,n)

A =

2 3 4 5 6

3 4 5 6 7

4 5 6 7 8

%(d)

m=4,n=4

m =

4

n =

4

A=sums(m,n)

A =

2 3 4 5

3 4 5 6

4 5 6 7

5 6 7 8

save Project0

type switches

function B=switches(A)

%This takes a matrix and swaps its rows to columns and columns to rows

[r,c] =size(A);%gets the size of Matrix A. r is rows, c is columns

C=zeros(c,r); %Need to rotate matrix so number of rows now become number of columns and visa versa

for i=1:c

for j=1:r

C(i,j) = A(j,i);

end

end

B=C;

end

%(c)

m=3,n=4

m =

3

n =

4

A=sums(m,n)

A =

2 3 4 5

3 4 5 6

4 5 6 7

switches(A)

ans =

2 3 4

3 4 5

4 5 6

5 6 7

%(d)

m=4,n=4

m =

4

n =

4

A=sums(m,n)

A =

2 3 4 5

3 4 5 6

4 5 6 7

5 6 7 8

switches(A)

ans =

2 3 4 5

3 4 5 6

4 5 6 7

5 6 7 8

%You can see that there is no change to (d) after doing the fuction switches

%This is becasue the number in the rows are in the same order as in the columns

%So you are just swapping the same numbers with each other

save Project0

diary close