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

size(A)

ans =

3 4

size(B)

ans =

3 2

size(X)

ans =

3 1

size(x)

ans =

1 4

size(y)

ans =

1 4

%Are the matrices x and y of the same size?

%Yes as a single line of horizontal matrix is equivalent to a single line

%of vertical matrix, a reshaping of one matrix to another.

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

%Display the value of matrix A at (row 1, column 3), equals to 8.

A(:,3)

ans =

8

3

5

%Display the values at the 3rd column of matrix A, equals to [8; 3; 5].

A(2,:)

ans =

0 -2 3 1

%Display the values at the 2nd row of matrix A, equals to [0 -2 3 1].

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

ans =

8 3

3 1

%Display the value of matrix A at (row 1 and 2, column 3 and 4), equals to

%[8 3; 3 1].

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

F =

5 2 -3 -1

4 3 -2 1

0 -1 6 -4

1 0 -2 3

% Add a 4th column to matrix F containing the values of [-1 1 -4 3].

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

%Changed the the values of matrix A at (row 1, column 2) to 1, at (row 1,

%column 4) to -3.

%Changed the the values of matrix A at (row 3, column 2) to 2, at (row 3,

%column 4) to -4.

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

F =

5 1 -3 -3

-2 -1 8 3

-3 -7 5 4

1 0 -2 3

%Replace the values of matrix F at (row 2 and 3, all columns) to the values

%of matrix A (row 1 and 3, all columns).

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

F =

1 5 -3 -3

-1 -2 8 3

-7 -3 5 4

0 1 -2 3

%Replace the values of matrix F at (all rows , columns 1 and 2) to the

%values of matrix A (all row, columns 2 and 1). Basically, the values in

%column 1 swap place with the values of column 2 for all rows in matrix F.

F(:,1)=y

F =

1 5 -3 -3

4 -2 8 3

6 -3 5 4

3 1 -2 3

%The command that will replace the first column of the matrix 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

%The command that switches rows 1, 2 in the matrix F.

[A B]

ans =

-2 -1 8 3 1 -2

0 -2 3 1 3 -1

-3 -7 5 4 1 0

%The matrix of A and B were combined, with the new matrix being 3x6 and

%matrix A being placed ahead, columns 1-4, followed by matrix B, columns

%5-6.

[B A]

ans =

1 -2 -2 -1 8 3

3 -1 0 -2 3 1

1 0 -3 -7 5 4

%The matrix of A and B were combined, with the new matrix being 3x6 and

%matrix B being placed ahead, columns 1-2, followed by matrix A, columns

%3-6.

[A x]

{\_Error using <a href="matlab:matlab.internal.language.introspective.

errorDocCallback('horzcat')" style="font-weight:bold">horzcat</a>

Dimensions of matrices being concatenated are not consistent.}\_

[A X]

ans =

-2 -1 8 3 -2

0 -2 3 1 3

-3 -7 5 4 5

%The matrix of A and X were combined, with the new matrix being 3x5 and

%matrix A being placed ahead, columns 1-4, followed by matrix X, columns 5.

[A ; y]

ans =

-2 -1 8 3

0 -2 3 1

-3 -7 5 4

1 4 6 3

%The matrix of A and y were combined, with the new matrix being 4x4 and

%matrix A being placed on top, rows 1-3, followed by matrix y, row 4.

[A ; x]

ans =

-2 -1 8 3

0 -2 3 1

-3 -7 5 4

1 2 3 4

%The matrix of A and x were combined, with the new matrix being 4x4 and

%matrix A being placed on top, rows 1-3, followed by matrix x row 4.

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

%Create a 5x5 Identity matrix.

zeros(3,4)

ans =

0 0 0 0

0 0 0 0

0 0 0 0

%Create a 3x4 Zero matrix.

zeros(2)

ans =

0 0

0 0

%Create a 2x2 Zero matrix.

ones(3,2)

ans =

1 1

1 1

1 1

%Create a 3x2 ones matrix, filled with only 1s.

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

%Create a 5x5 ones matrix, filled with only 1s.

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

%Create a 5x5 matrix, with the values of [1 2 5 6 7] going diagonally

%across. (1 being at the top left hand corner and 7 being at the

%bottom right hand corner.

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

%Create a 6x6 matrix, with a zeros row and column on the outer border

%and the values of [1 2 5 6 7] going diagonally

%across. (1 being at the top left hand corner and 7 being at the

%bottom right hand corner.

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

%Create a 7x7 matrix, with a two zeros row and column on the inner border

%and the values of [1 2 5 6 7] going diagonally after two zeros column

%(1 being at the top left hand corner and 7 being at the

%bottom right hand corner above the two zeros row.

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

%Display matrix A and its diagonal values in a 3x1 matrix and as a 3x3

%matrix with its original position and all other non-diagonal values

%being zero out.

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

%Display a 5x5 matrix with integers 1 through n^2 with equal row and

%column 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>

%Display "magic" function reference, explains what "magic" is.

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

%Displau the Hilbert matrix of order 5.

help hilb

<strong>hilb</strong> Hilbert matrix.

<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.

See INVHILB for the exact inverse.

<strong>hilb</strong>(N,CLASSNAME) produces a matrix of class CLASSNAME.

CLASSNAME must be either 'single' or 'double' (the default).

This is also a good example of efficient MATLAB programming

style where conventional FOR or DO loops are replaced by

vectorized statements.

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

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

%Display "hilb" function reference, explains what "hilb" is.

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

V1=1:7

V1 =

1 2 3 4 5 6 7

%Display a 1x7 vector matrix with values in order of 1 to 7.

V2=2:0.5:6.5

V2 =

Columns 1 through 7

2.0000 2.5000 3.0000 3.5000 4.0000 4.5000 5.0000

Columns 8 through 10

5.5000 6.0000 6.5000

%Display a 1x10 vector matrix with values in order of 2 to 6.5 in

%increment of 0.5.

V3=3:-1:-5

V3 =

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

%Display a 1x9 vector matrix with values in order of 3 to 0 to -5 in

%in reverse order/ decreasing.

V4=-5:1

V4 =

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

%command needed to create each of the following vectors in colon

%notation: V4=[-5 - 4 - 3 -2 -1 0 1]

V5=10:-3:-2

V5 =

10 7 4 1 -2

%command needed to create each of the following vectors in colon

%notation: V5=[10 7 4 1 -2]

V6=5:-0.5:2

V6 =

5.0000 4.5000 4.0000 3.5000 3.0000 2.5000 2.0000

%command needed to create each of the following vectors in colon

%notation: V6=[5 4.5 4 3.5 3 2.5 2]

V7=0:0.4:4

V7 =

Columns 1 through 7

0 0.4000 0.8000 1.2000 1.6000 2.0000 2.4000

Columns 8 through 11

2.8000 3.2000 3.6000 4.0000

%command needed to create each of the following vectors in colon

%notation: the numbers from 0 to 4 spread 0.4 apart.

C;

% the command was executed, but the result was not displayed

%on the screen.

C

C =

1 0 0

0 1 0

0 0 1

%The values of C was display on screen.

R=434.1452

R =

434.1452

%Set value of R to 434.1452.

format long, R

R =

4.341452000000000e+02

%Place the value of R in scientific notation with extended decimal

%lenth.

format short, R

R =

434.1452

%Place the value of R back to its original input form.

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.

}\_

%The dimension of matrix B (3x2) does not match that of matrix A (3x4)

%resulting in an error.

B, E, B-2\*E

B =

1 -2

3 -1

1 0

E =

1 1 1

1 1 1

{\_Matrix dimensions must agree.

}\_

%The dimension of matrix B (3x2) does not match that of matrix E (2x3)

%resulting in an error.

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 6 9 7

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.

}\_

%Attempts multiplies arrays A and D element by element and returns result.

%The dimension of matrix A does not match that of matrix D

%resulting in an error.

A\*A

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

Inner matrix dimensions must agree.

}\_

%This operation is assuming that you will perform a matrix multiplication.

% Which causes an error. What I actually want to do is an element-

%by-element operation.

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

G =

4 2 1

3 1 6

7 7 8

G^2, A^2

ans =

29 17 24

57 49 57

105 77 113

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

One argument must be a square matrix and the other must be a scalar. Use

POWER (.^) for elementwise power.

}\_

% The command is equivalent to G^2 is G\*G.

%A^2 produces an error message because A is not a square matrix.

%The command that calculates element-wise power 2 of A is A.^2.

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

%Returns an n-by-n matrix of random numbers between 0-1.

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

%Returns an 3x4 matrix of random numbers between 0-1.

randi(100,2)

ans =

66 71

18 4

%Returns an 2x2 matrix of random numbers between 0-100.

randi(10,2,4)

ans =

3 1 7 10

1 9 4 1

%Returns an 2x4 matrix of random numbers between 0-10.

randi([10 40],2,4)

ans =

23 33 15 23

21 34 25 30

%Returns an 2x4 matrix of random numbers between 10-40.

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

4 + (10-4).\*rand(2,3)

ans =

4.4551 7.1848 9.6041

4.3237 8.6750 4.7794

randi([40 90],2,3)

ans =

69 40 48

63 57 80

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

type sums

function A=sums(m,n)

% This is the function which adds

% matrices A and B. It duplicates the MATLAB

% function A+B.

[a,b]=size(m);

[k,p]=size(n);

if a==k && b==p,

for i=1:a

for j=1:b

A(i,j)=m(i,j)+n(i,j);

end

end

else

disp('Error in using sums')

A=[];

end

end

% (a)

m=6.5, n=4

m =

6.5000

n =

4

A=sums(m,n)

A =

10.5000

% (b)

m = 5, n = 1.6

m =

5

n =

1.6000

A=sums(m,n)

A =

6.6000

% (c)

m = 3, n = 4

m =

3

n =

4

A=sums(m,n)

A =

7

% (d)

m = 3, n = 3

m =

3

n =

3

A=sums(m,n)

A =

6

type switches

function B = switches(A)

%This function switches rows with columns in a matrix A.

%B(j,i)=A(i,j) (i = 1 : m, j = 1 : n);

[r c] = size(A);

B = zeros(c,r);

for i = 1:r

for j = 1:c

B(j,i) = A(i,j)

end

end

end

%(c) switches

A=[3 4]

A =

3 4

B = switches(A)

B =

3

0

B =

3

4

%(d) switches

A=[3 3]

A =

3 3

B = switches(A)

B =

3

0

B =

3

3

%The function switches rows with columns in a matrix the

%output in part (d).

%(c) switches

A =[3 4]

A =

3 4

A.'

ans =

3

4

%(d) switches

A=[3 3]

A =

3 3

A.'

ans =

3

3

diary off