Matlab

Section B: Useful Functions in Matlab

**all**
For vectors, all(V) returns logical 1 (TRUE) if none of the elements of the vector are zero. Otherwise it returns logical 0 (FALSE). For matrices, ALL(X) operates on the columns of X, returning a row vector of logical 1’s and 0’s. For N-D arrays, ALL(X) operates on the first non-singleton dimension.

all(X,DIM) works down the dimension DIM. For example, ALL(X,1) works down the first dimension (the rows) of X.

**any**
B = any(A) tests whether any of the elements along various dimensions of an array is a nonzero number or is logical 1 (true).

any ignores entries that are NaN (Not a Number).

If A is a vector, any(A) returns logical 1 (true) if any of the elements of A is a nonzero number or is logical 1 (true) and returns logical 0 (false) if all the elements are zero.

If A is a matrix, any(A) treats the columns of A as vectors, returning a row vector of logical 1’s and 0’s.

**ceil**
B = ceil(A) rounds the elements of A to the nearest integers greater than or equal to A. For complex A, the imaginary and real parts are rounded independently.

**char**
S = char(A) converts the array A that contains nonnegative integers representing character codes into a MATLAB character array (the first codes are ASCII). The actual characters displayed depends on the 121-character encoding scheme for a given font. The result for any elements of A outside the range from 0 to 65535 is not defined (and may vary from platform to platform). Use DOUBLE to convert a character array into its numeric codes.

**floor**
B = floor(A) rounds the elements of A to the nearest integers less than or equal to A. For complex A, the imaginary and real parts are rounded independently.

**isempty**
logical 0 (false) 1 (true) if A is an empty array and TF = isempty(A) returns logical one dimension of size zero, for example, 0-by- otherwise. An empty array has at least 0 or 0-by-5.
length
n = length(X) returns the size of the longest dimension of X. If X is a vector, this is the same as its length.

find
ind = find(X) locates all nonzero elements of array X, and returns the linear indices of those elements in vector ind. If X is a row vector, then ind is a row vector; otherwise, ind is a column vector. If X contains no nonzero elements or is an empty array, then ind is an empty array.
ind = find(X, k) or ind = find(X, k, 'first') returns at most the first k indices corresponding to the nonzero entries of X. k must be a positive integer, but it can be of any numeric data type. ind = find(X, k, 'last') returns at most the last k indices corresponding to the nonzero entries of X.
[row, col] = find(X, ...) returns the row and column indices of the nonzero entries in the matrix X.
This syntax is especially useful when working with sparse matrices. If X is an N dimensional array with N > 2, col contains linear indices for the columns.

max
C = max(A) returns the largest elements along different dimensions of an array.
If A is a vector, max(A) returns the largest element in A.
If A is a matrix, max(A) treats the columns of A as vectors, returning a row vector containing the maximum element from each column.
If A is a multidimensional array, max(A) treats the values along the first non-singleton dimension as vectors, returning the maximum value of each vector.

min
C = min(A) returns the smallest elements along different dimensions of an array.
If A is a vector, min(A) returns the smallest element in A.
If A is a matrix, min(A) treats the columns of A as vectors, returning a row vector containing the minimum element from each column.
If A is a multidimensional array, min operates along the first nonsingleton dimension. C = min(A,[],dim) returns the smallest elements along the dimension of A specified by scalar dim. For example, min(A,[],1) produces the minimum values along the first dimension (the rows) of A.
mod
M = mod(X,Y) if Y ~= 0, returns X - n.*Y where n = floor(X./Y). If Y is not an integer and the quotient X./Y is within roundoff error of an integer, then n is that integer.
The inputs X and Y must be real arrays of the same size, or real scalars.
Example:
X = [23 24 25 26]
M = mod(X, 3)
Then M gets the array [2 0 1 2]

ones
Y = ones(n) returns an n-by-n matrix of 1s. An error message appears if n is not a scalar.
Y = ones(m,n) or Y = ones([m n]) returns an m-by-n matrix of ones.
Y = ones(m,n,p,...) or Y = ones([m n p ...]) returns an m-by-n-by-p-by-... array of 1s.

repmat
B = repmat(A,m,n) creates a large matrix B consisting of an m-by-n tiling of copies of A. The size of B is [size(A,1)*m, size(A,2)*n].
The statement repmat(A,n) creates an n-by-n tiling.
B = repmat(A,[m n]) accomplishes the same result as repmat(A,m,n).
B = repmat(A,[m n p...]) produces a multidimensional array B composed of copies of A.
The size of B is [size(A,1)*m, size(A,2)*n, size(A,3)*p, ...].

reshape
B = reshape(X,M,N) returns the M-by-N matrix whose elements are taken columnwise from X. An error results if X does not have M*N elements.

size
d = size(X) returns the sizes of each dimension of array X in a vector d with ndims(X) elements. If X is a scalar, which MATLAB regards as a 1-by-1 array, size(X) returns the vector [1 1].
[m,n] = size(X) returns the size of matrix X in separate variables m and n.
m = size(X,dim) returns the size of the dimension of X specified by scalar dim.
[d1,d2,d3,...,dn] = size(X), for n >1, returns the sizes of the dimensions of the array X in the variables d1,d2,d3,...,dn, provided the number of output arguments n equals ndims(X).
sum
B = sum(A) returns sums along different dimensions of an array.
If A is a vector, sum(A) returns the sum of the elements.
If A is a matrix, sum(A) treats the columns of A as vectors, returning a row vector of
the sums of each column.
If A is a multidimensional array, sum(A) treats the values along the first non-singleton
dimension as vectors, returning an array of row vectors.
B = sum(A,dim) sums along the dimension of A specified by scalar dim. The dim
input is an integer value from 1 to N, where N is the number of dimensions in A. Set
dim to 1 to compute the sum of each column, 2 to sum rows, etc.

zeros
B = zeros(n) returns an n-by-n matrix of zeros. An error message appears if n is not a
scalar.
B = zeros(m,n) or B = zeros([m n]) returns an m-by-n matrix of zeros.
B = zeros(m,n,p,...) or B = zeros([m n p ...]) returns an m-by-n-by-p-by-... array of
zeros.

isequal
tf = isequal(A,B) returns logical 1 (true) if A and B are equivalent; otherwise, it returns
logical 0 (false).

ismember
Lia = ismember(A,B) returns an array containing logical 1 (true) where the data in A is
found in B. Elsewhere, the array contains logical 0 (false).

sort
B = sort(A) sorts the elements of A in ascending order.
• If A is a vector, then sort(A) sorts the vector elements.
• If A is a matrix, then sort(A) treats the columns of A as vectors and sorts each
column.
B = sort(A,dim) returns the sorted elements of A along dimension dim. For example,
if A is a matrix, then sort(A,2) sorts the elements of each row.