Contents

Octave Functions

Matrix / Vector related functions

transpose(a)	This function returns transpose of matrix `a' passed to it as argument. It can also be called with operator ('), for example a'
inverse(a)	Returns inverse of matrix `a' given as argument
inv(a)	Same as inverse(a). Returns inverse of matrix `a' given as argument
det(a)	Returns determinant of matrix `a' given as argument
linspace(a,b,c)	Returns row vector contains `c' uniformly distributed values between `a' and `b' given as arguments. If a<b then values are arranged in descending order. Argument c is optional and by default 100 values are generated between a and b in case third argument is not present.
rand(a,b)	Returns matrix of dimensions axb with random numbers as elements of matrix. If argument b is omitted then it returns square matrix of dimension axa.
eye(a,b)	Returns indentity matrix of dimensions axb. If argument b is not supplied then returns square identity matrix of dimensions axa.

NA(a,b)	NA() function can be used to generate matrix of special Not Available('NA') data type. The returned matrix is of dimensions axb with all values of matrix of type NA. In case we dont specify argument b then returned matrix is square matrix of dimensions axa.
ndims(a)	Returns number of dimensions of argument `a' passed to it. Usually all values are stored as matrices and hence minimum 2 is printed when we call ndims()
rows()	Returns number of rows in matrix / array passed as argument.
columns()	Returns number of columns in matrix / array passed as argument
numel()	Returns number of elements in matrix / array passed as argument
length()	Returns length of the object. For strings length would mean number of characters in string. For matrices length returns number of rows or columns whichever is greater.
size()	Returns a 1x2 matrix with number of rows and columns. If we call size() with another argument, say 1 or 2, then it returns number of elements in that dimension of matrix.
zeros()	Create a matrix of size specified by arguments with all values as 0.
true()	Returns matrix of true values. The dimensions of matrix depend on arguments supplied to this function.
false()	Returns matrix of false values. The dimensions of matrix depend on arguments supplied to this function.

Input/Output functions

plot()	Plot function can be used to draw graph from values of a vector. • If we call plot with only one argument of dimension Nx1 then plot with draw graph with index of values on X-axis and values of argument on Y-axis. • In case we call plot with only one argument of dimension N*M then plot with draw M graphs of each N*1 column vector present in the argument, each with different color on same drawing area. • If we call plot with two arguments each with dimensions 1xN then plot will draw graph with values of first argument as X-values and values of second argument as Y-values for points of graph.

	• If we call plot with three arguments each with dimensions 1xN then plot will draw two graphs First graph with values of first argument as X-values and values of second argument as Y-values and second graph with values of first argument as X-values and values of third argument as Y-values. • If we call plot with four arguments each with dimensions 1xN then plot will draw two graphs. First graph with values of first argument as X-values and values of second argument as Y-values and second graph with values of third argument as X-values and values of fourth argument as Y-values.
printf()	It is very similar to C programming language, printf() function. It can be used to generate formatted output.
output_max_field_width()	Both functions set / display number of digits to display after decimal point in output
output_precision()	Same as output_max_field_width ()
split_long_rows()	Used to specify / find whether octave should print matrices with many columns split among various lines or not.
struct_levels_to_print()	Can be used to query or set the internal variable that specifies the number of structure levels to display.

System functions

exit()	It ends octave with exit code passed as argument to function.
quit()	Same as exit, it ends octave with exit code passed as argument to function.

Data type related functions

typeinfo()	If called without any argument it prints list of all currently loaded data types. If called with argument it returns data type of argument.
class()	Returns the class of expression passed as argument. The return value is in string format.

cast()	cast() function can be used to covert value of one class to other. That is it can be used to convert double into char and vice-versa. It is important to understand that when we convert char to double we will get ascii value of char and not parsed decimal value. Hence char 'a' when coverted to double will give us value 97. Similarly char '0' when converted to double gives us value '48'.
sizeof()	Returns number of bytes required to store value in memory
size_equal()	Returns true if size of all arguments passed to this function are same.
int8()	Converts argument to 8-bit integer type.
uint8()	Converts argument to unsigned 8-bit integer type.
int16()	Converts argument to 16-bit integer type.
uint16()	Converts argument to unsigned 16-bit integer type
int32()	Converts argument to signed 32-bit integer type
uint32()	Converts argument to unsigned 32-bit integer type
int64()	Converts argument to signed 64-bit integer type
uint64()	Converts argument to unsigned 64-bit integer type
intmax()	Expects a integer class type as argument (ex "int8", "int16", etc.) and returns maximum possible integer that can be represented by that type.
intmin()	Expects a integer class type as argument (ex "int8", "int16", etc.) and returns minimum possible integer that can be represented by that type.
logical()	Converts given matrix to its logical equivalent true or false matrix. Only 0 is treated as false, rest all other values are converted to true.
char()	Converts integer argument supplied to character value. If argument is matrix then result is array of strings

Testing functions

isinteger()	Returns true if object given as argument is integer. Note that normal constants 4,8,14 etc. are not integers as by default constants are treated as double.
isempty()	isempty() returns 1 if at least one dimension of argument is of length 0 otherwise it returns 1.
isna()	isna() functions takes one matrix argument and returns another matrix of same dimension. If any value of matrix passed as argument is data type Not Available('NA') then corresponding values in returned matrix are 1 and all other values are 0. Note that even Inf and NaN are not treated as NA. Hence only when original value is NA, 1 is returned in its place.
isa()	Function isa() expects two arguments and checks whether first argument belongs to class described by second argument. Second argument must be class name in string format. If the object belongs to class name passed as second argument then return value is integer '1' else return value is integer '0'
isnumeric()	Returns true if argument supplied is numeric
isreal()	Returns true if argument supplied is real number (not complex number)
iscomplex()	Returns true if argument supplied is complex number
ismatrix()	Returns true if argument supplied is matrix
isvector()	Returns true if argument supplied is vector
isscalar()	Returns true if argument supplied is scalar, not vector
issquare()	Returns dimension (width or height) of matrix if it is square matrix, else it returns 0
issymmetric()	Returns zero if matrix is not symmetric, non-zero value other wise.
islogical()	Returns true is argument is logical, false otherwise
isprime()	Returns true if argument is prime, false otherwise
isstruct()	Returns true if argument is structure

isfield(a,b)	Returns true if a is structure containing b as element. a must be structure and b must be string
isvarname(a)	Returns true if string `a' is valid variable name
isglobal(a)	Returns true if variable name `a' is global
exist(a,b)	Returns non-zero value if name specified by string `a' is already used as function / variable / file name in local path. Optional parameter `b' can take values `"var"', `"builtin"', `"file"' and `"dir"' in case we want to restrict search to specific domain. Use `help exist' to understand difference between different non-zero return values of function.
isequal()	Returns true if all arguments passed to this function are equal.
isequalwithequalnans()	Returns true if all arugments passed to this function are equal. For purpose of comparison this function would treat NaN as equal to NaN.

String functions

strcmp(a,b)	Compares strings a and b. **Unline C strcmp, this function returns 1 if strings are same and 0 otherwise**
strcmpi(a,b)	Compares strings a and b ignoring case. **Unline C strcasecmp, this function returns 1 if strings are same and 0 otherwise**
strncmp(a,b,N)	Compares first N characters of a and b. **Unline C strncmp, this function returns 1 if strings are same and 0 otherwise**
strncmpi(a,b,N)	Compares first N characters of a and b ignoring case. **Unline C strncasecmp, this function returns 1 if strings are same and 0 otherwise**
index(a,b)	Find position of first occurrence of b in a.
rindex(a,b)	Find position of last occurrence b in a.
strrep(s,x,y)	Replaces all occurrences of the substring x of the string s with the string y.
substr(s,offset,len)	Return the substring of s which starts at character number offset and is len characters long. If offset is negative, extraction starts that far from the end of the string. If len is

	omitted, the substring extends to the end of S.

Conversion functions

bin2dec(s)	Return the decimal number corresponding to the binary number stored in the string s.
dec2bin(n,len)	Return a binary number corresponding the non-negative decimal number n, as a string of ones and zeros. The optional second argument, len, specifies the minimum number of digits in the result.
dec2hex(n,len)	Return the hexadecimal string corresponding to the non-negative integer n. The optional second argument, len, specifies the minimum number of digits in the result.
hex2dec(s)	Return the integer corresponding to the hexadecimal number stored in the string s.
dec2base(n,b,len)	Return a string of symbols in base b corresponding to the the non-negative integer n. The optional third argument, len, specifies the minimum number of digits in the result. The optional third argument, len, specifies the minimum number of digits in the result. (That is, if "dec2base (123, 3) => "11120" then dec2base (123, "aei") => "eeeia")
base2dec(s,b)	Convert s from a string of digits of base b into an integer. If b is a string, the characters of b are used as the symbols for the digits of s. Space (' ') may not be used as a symbol. (That is, if base2dec ("11120", 3) => 123 then base2dec ("yyyzx", "xyz") => 123)
str2double(str,cdelim,rdelim,ddelim)	Convert strings into numeric values. The parameters cdelim, rdelim, and ddelim are optional column, row, and decimal delimiters. The default row-delimiters are newline, carriage return and semicolon (ASCII 10, 13 and 59). The default column-delimiters are tab, space and comma (ASCII 9, 32, and 44). The default decimal delimiter is `.' (ASCII 46).
str2num(s)	Convert the string s to a number.
toascii(s)	Return ASCII representation of s in a matrix. For example, toascii ("ASCII") => [65, 83, 67, 73, 73]

tolower(s)	Return a copy of the string s, with each upper-case character replaced by the corresponding lower-case one.
toupper(s)	Return a copy of the string s, with each lower-case character replaced by the corresponding upper-case one.
do_string_escapes(string)	Convert special characters in string to their escaped forms. This is like PHP stripslashes() function.
undo_string_escapes(s)	Converts special characters in strings back to their escaped forms. This is like PHP addslashes() function.
ind2sub()	Converts index of multi-dimensional matrix into linear index. Use ``help ind2sub`' to understand the usage / purpose of this function properly.
sub2ind()	Converts linear index of multi-dimensional matrix into multi-dimensional index. Use ``help sub2ind`' for detailed help on how to use this function.

Character class functions

Below mentioned functions operate on string arrays and return matrices of zeros and ones. Elements that are nonzero indicate that the condition was true for the corresponding character in the string array.

isalnum(s)	Return 1 for characters that are letters or digits. (isalpha(s) or isdigit(s) is true)
isalpha(s)	Return true for characters that are letters (isupper(s) or islower(s) is true)
isletter(s)	Same as isalpha(s)
isascii(s)	Return 1 for characters that are ASCII
iscntrl(s)	Return 1 for control characters.
isdigit(s)	Return 1 for characters that are decimal digits.
isgraph(s)	Return 1 for printable characters (but not the space character).
islower(s)	Return 1 for characters that are lower case letters.

isprint(s)	Return 1 for printable characters (including the space character)
ispunct(s)	Return 1 for punctuation characters.
isspace(s)	Return 1 for whitespace characters (space, formfeed, newline, carriage return, tab, and vertical tab).
isupper(s)	Return 1 for upper case letters.
isxdigit(s)	Return 1 for characters that are hexadecimal digits.

Cell array retlated functions

fieldnames()	Returns a cell array of strings naming the elements of the structure passed as argument. Argument passed to this function must be a structure.