

## Embedded MATLAB Run-Time Function Library -- Alphabetical List

This topic lists functions in the Embedded MATLAB run-time library in alphabetical order. Each entry includes a link to the help for the equivalent MATLAB or Fixed-Point Toolbox function along with any limitations that apply to it. See also [Embedded MATLAB Run-Time Library -- Categorical List](#)

Function	Product	Remarks/Limitations
<a href="#">abs</a>	MATLAB	—
<a href="#">abs</a>	Fixed-Point Toolbox	—
<a href="#">acos</a>	MATLAB	<ul style="list-style-type: none"> <li>Returns NaN when the input value <math>x</math> is real and the output should be complex. To get the result, make the input value complex by using <code>complex(x)</code>.</li> </ul>
<a href="#">acosd</a>	MATLAB	—
<a href="#">acosh</a>	MATLAB	<ul style="list-style-type: none"> <li>Returns NaN when the input value <math>x</math> is real and the output should be complex. To get the result, make the input value complex by using <code>complex(x)</code>.</li> </ul>
<a href="#">acot</a>	MATLAB	—
<a href="#">acotd</a>	MATLAB	—
<a href="#">acoth</a>	MATLAB	—
<a href="#">acsc</a>	MATLAB	—
<a href="#">acscd</a>	MATLAB	—
<a href="#">acsch</a>	MATLAB	—
<a href="#">all</a>	MATLAB	—
<a href="#">all</a>	Fixed-Point Toolbox	—
<a href="#">and</a>	MATLAB	—
<a href="#">angle</a>	MATLAB	—
<a href="#">any</a>	MATLAB	—
<a href="#">any</a>	Fixed-Point Toolbox	—
<a href="#">asec</a>	MATLAB	—
<a href="#">asecd</a>	MATLAB	—
<a href="#">asech</a>	MATLAB	—



<a href="#">asin</a>	MATLAB	<ul style="list-style-type: none"> <li>• Returns NaN when the input value <math>x</math> is real and the output should be complex. To get the complex result, make the input value complex by using <code>complex(x)</code>.</li> </ul>
<a href="#">asind</a>	MATLAB	—
<a href="#">asinh</a>	MATLAB	—
<a href="#">atan</a>	MATLAB	—
<a href="#">atan2</a>	MATLAB	—
<a href="#">atand</a>	MATLAB	—
<a href="#">atanh</a>	MATLAB	<ul style="list-style-type: none"> <li>• Returns NaN when the input value <math>x</math> is real and the output should be complex. To get the complex result, make the input value complex by using <code>complex(x)</code>.</li> </ul>
<a href="#">bitand</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not support floating point inputs. The arguments must belong to an integer class.</li> </ul>
<a href="#">bitand</a>	Fixed-Point Toolbox	—
<a href="#">bitcmp</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not support floating point input for the first argument. The first argument must belong to an integer class.</li> </ul>
<a href="#">bitcmp</a>	Fixed-Point Toolbox	—
<a href="#">bitget</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not support floating point input for the first argument. The first argument must belong to an integer class.</li> </ul>
<a href="#">bitget</a>	Fixed-Point Toolbox	—
<a href="#">bitor</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not support floating point inputs. The arguments must belong to an integer class.</li> </ul>
<a href="#">bitor</a>	Fixed-Point Toolbox	—
<a href="#">bitset</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not support floating point input for the first argument. The first argument must belong to an integer class.</li> </ul>
<a href="#">bitset</a>	Fixed-Point Toolbox	—
<a href="#">bitshift</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not support floating point input for the first argument. The first argument must belong to an integer class.</li> </ul>

al, but e complex passing
al, but e complex passing
he ss.
the first ng to an
the first ng to an
he ss.
the first ng to an
the first ng to an

<a href="#">bitshift</a>	Fixed-Point Toolbox	—
<a href="#">bitxor</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not support floating point inputs. TI arguments must belong to an integer class</li> </ul>
<a href="#">bitxor</a>	Fixed-Point Toolbox	—
<a href="#">cast</a>	MATLAB	
<a href="#">ceil</a>	MATLAB	—
<a href="#">char</a>	MATLAB	—
<a href="#">chol</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not allow two output arguments</li> </ul>
<a href="#">class</a>	MATLAB	—
<a href="#">compan</a>	MATLAB	—
<a href="#">complex</a>	MATLAB	—
<a href="#">complex</a>	Fixed-Point Toolbox	—
<a href="#">conj</a>	MATLAB	—
<a href="#">conj</a>	Fixed-Point Toolbox	—
<a href="#">conv</a>	MATLAB	—
<a href="#">cos</a>	MATLAB	—
<a href="#">cosd</a>	MATLAB	—
<a href="#">cosh</a>	MATLAB	—
<a href="#">cot</a>	MATLAB	—
<a href="#">cotd</a>	MATLAB	—
<a href="#">coth</a>	MATLAB	—
<a href="#">cross</a>	MATLAB	<ul style="list-style-type: none"> <li>• If supplied, <code>dim</code> must be a constant</li> </ul>
<a href="#">csc</a>	MATLAB	—
<a href="#">cscd</a>	MATLAB	—
<a href="#">csch</a>	MATLAB	—
<a href="#">ctranspose</a>	MATLAB	—
<a href="#">ctranspose</a>	Fixed-Point Toolbox	—
<a href="#">cumprod</a>	MATLAB	—
<a href="#">cumsum</a>	MATLAB	—

he  
ss.

<a href="#">diag</a>	MATLAB	<ul style="list-style-type: none"> <li>• If supplied, <math>k</math> must be a real and scalar integer value</li> </ul>
<a href="#">diff</a>	MATLAB	<ul style="list-style-type: none"> <li>• If applied, <math>n</math> and <math>\text{dim}</math> must be constants</li> </ul>
<a href="#">disp</a>	Fixed-Point Toolbox	—
<a href="#">divide</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Any non-<math>\text{fi}</math> input must be constant; that value must be known at compile time so it can be cast to a <math>\text{fi}</math></li> <li>• Complex and imaginary divisors are not supported</li> </ul>
<a href="#">dot</a>	MATLAB	—
<a href="#">double</a>	MATLAB	—
<a href="#">double</a>	Fixed-Point Toolbox	—
<a href="#">eig</a>	MATLAB	<ul style="list-style-type: none"> <li>• QZ algorithm used in all cases. Consequently, the standard eigenvalue problem &amp; identical results will be similar to those obtained using MATLAB:  <math display="block">[V,D] = \text{eig}(A, \text{eye}(\text{size}(A)))</math> </li> <li>• However, <math>V</math> may represent a different basis of eigenvectors, and the eigenvalues in <math>D</math> may not be in the same order.</li> <li>• Options 'balance', 'nobalance', and 'chol' are not yet supported.</li> <li>• Outputs are always of complex type.</li> </ul>
<a href="#">end</a>	Fixed-Point Toolbox	—
<a href="#">eps</a>	MATLAB	—
<a href="#">eps</a>	Fixed-Point Toolbox	—
<a href="#">eq</a>	MATLAB	—
<a href="#">eq</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Not supported for fixed-point signals with biases</li> </ul>
<a href="#">exp</a>	MATLAB	—
<a href="#">eye</a>	MATLAB	<ul style="list-style-type: none"> <li>• Dimensions must be real, non-negative, integer constants</li> </ul>
<a href="#">factorial</a>	MATLAB	—
<a href="#">false</a>	MATLAB	Dimensions must be real, non-negative, integer constants
<a href="#">fft</a>	MATLAB	<ul style="list-style-type: none"> <li>• Length of input vector must be a power of 2</li> </ul>

ninteger
is, its that it
supported
ently, for tity), ising the
, 'qz' )
sis of ay not be
nd
n different
integer
' constants
of 2



<a href="#">fftshift</a>	MATLAB	—
<a href="#">fi</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Use to create a fixed-point constant or variable in Embedded MATLAB</li> <li>• The syntax <code>fi('PropertyName',PropertyValu</code> is not supported. To use property name/value pairs, you must first specify the value of the <code>fi</code> object as in <code>fi(v,'PropertyName',PropertyVal</code></li> <li>• Works for constant input values only; the value of the input must be known at compile time</li> <li>• <code>Numerictype</code> information must be available for nonfixed-point Simulink inputs</li> </ul>
<a href="#">filter</a>	MATLAB	<ul style="list-style-type: none"> <li>• Results might differ from MATLAB if the input signal contains NaNs</li> </ul>
<a href="#">fimath</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Fixed-point signals coming in to an Embedded MATLAB Function block from Simulink are automatically assigned the default <code>fimath</code> object</li> <li>• <code>fimath</code> objects in Embedded MATLAB</li> </ul>
<a href="#">fix</a>	MATLAB	—
<a href="#">fliplr</a>	MATLAB	—
<a href="#">flipud</a>	MATLAB	—
<a href="#">floor</a>	MATLAB	—
<a href="#">freqspace</a>	MATLAB	—
<a href="#">ge</a>	MATLAB	—
<a href="#">ge</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Not supported for fixed-point signals with biases</li> </ul>
<a href="#">gt</a>	MATLAB	—
<a href="#">gt</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Not supported for fixed-point signals with biases</li> </ul>
<a href="#">hilb</a>	MATLAB	—
<a href="#">horzcat</a>	Fixed-Point Toolbox	—
<a href="#">idivide</a>	MATLAB	<ul style="list-style-type: none"> <li>• <code>opt</code> string must be in lower case</li> <li>• For efficient generated code, MATLAB divide-by-zero rules are supported only for the 'round' option</li> </ul>
<a href="#">ifft</a>	MATLAB	<ul style="list-style-type: none"> <li>• Length of input vector must be a power of 2</li> <li>• Output of <code>ifft</code> block is always complex</li> </ul>

variable in
ue...) property lue $v$ of
alue...) at is, the pile time ilable for
input
edded re
n different
n different
for the
of 2

<a href="#">imag</a>	MATLAB	—
<a href="#">imag</a>	Fixed-Point Toolbox	—
<a href="#">ind2sub</a>	MATLAB	<ul style="list-style-type: none"> <li>• No support for N-dimensional matrices. S vector must have exactly two elements.</li> </ul>
<a href="#">inf</a>	MATLAB	<ul style="list-style-type: none"> <li>• Dimensions must be real, non-negative, constants.</li> </ul>
<a href="#">int8</a> , <a href="#">int16</a> , <a href="#">int32</a>	MATLAB	—
<a href="#">int8</a> , <a href="#">int16</a> , <a href="#">int32</a>	Fixed-Point Toolbox	—
<a href="#">interp1</a>	MATLAB	<ul style="list-style-type: none"> <li>• Supports only linear and nearest interpolation methods</li> <li>• Does not handle evenly spaced x indices separately</li> <li>• x must be strictly monotonically increasing or strictly monotonically decreasing; does not support NaN indices</li> </ul>
<a href="#">interp1q</a> , see <a href="#">interp1</a>	MATLAB	<ul style="list-style-type: none"> <li>• x must be strictly monotonically increasing or strictly monotonically decreasing; does not support NaN indices</li> </ul>
<a href="#">intmax</a>	MATLAB	
<a href="#">intmin</a>	MATLAB	
<a href="#">inv</a>	MATLAB	—
<a href="#">invhilb</a>	MATLAB	—
<a href="#">isa</a>	MATLAB	—
<a href="#">ischar</a>	MATLAB	—
<a href="#">iscolumn</a>	Fixed-Point Toolbox	—
<a href="#">isempty</a>	MATLAB	—
<a href="#">isempty</a>	Fixed-Point Toolbox	—
<a href="#">isequal</a>	MATLAB	<ul style="list-style-type: none"> <li>• Supports only two arguments.</li> </ul>
<a href="#">isfi</a>	Fixed-Point Toolbox	—
<a href="#">isfimath</a>	Fixed-Point Toolbox	—
<a href="#">isfinite</a>	MATLAB	—



<a href="#">isfinite</a>	Fixed-Point Toolbox	—
<a href="#">isfloat</a>	MATLAB	—
<a href="#">isinf</a>	MATLAB	—
<a href="#">isinf</a>	Fixed-Point Toolbox	—
<a href="#">isinteger</a>	MATLAB	—
<a href="#">islogical</a>	MATLAB	—
<a href="#">isnan</a>	MATLAB	—
<a href="#">isnan</a>	Fixed-Point Toolbox	—
<a href="#">isnumeric</a>	MATLAB	—
<a href="#">isnumeric</a>	Fixed-Point Toolbox	—
<a href="#">isnumerictype</a>	Fixed-Point Toolbox	—
<a href="#">isreal</a>	MATLAB	—
<a href="#">isreal</a>	Fixed-Point Toolbox	—
<a href="#">isrow</a>	Fixed-Point Toolbox	—
<a href="#">isscalar</a>	MATLAB	—
<a href="#">isscalar</a>	Fixed-Point Toolbox	—
<a href="#">issigned</a>	Fixed-Point Toolbox	—
<a href="#">isvector</a>	MATLAB	—
<a href="#">isvector</a>	Fixed-Point Toolbox	—
<a href="#">ldivide</a>	MATLAB	—
<a href="#">le</a>	MATLAB	—
<a href="#">le</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Not supported for fixed-point signals with biases</li> </ul>
<a href="#">length</a>	MATLAB	—
<a href="#">length</a>	Fixed-Point Toolbox	—

n different

<a href="#">linspace</a>	MATLAB	<ul style="list-style-type: none"> <li>• Number of points <math>N</math> must be a constant, positive, real, and integer valued</li> </ul>
<a href="#">log</a>	MATLAB	<ul style="list-style-type: none"> <li>• Returns NaN when the input value <math>x</math> is real; the output should be complex. To get the result, make the input value complex by using <code>complex(x)</code>.</li> </ul>
<a href="#">logical</a>	MATLAB	—
<a href="#">logical</a>	Fixed-Point Toolbox	—
<a href="#">logspace</a>	MATLAB	—
<a href="#">lowerbound</a>	Fixed-Point Toolbox	—
<a href="#">lsb</a>	Fixed-Point Toolbox	—
<a href="#">lt</a>	MATLAB	—
<a href="#">lt</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Not supported for fixed-point signals with biases</li> </ul>
<a href="#">lu</a>	MATLAB	—
<a href="#">magic</a>	MATLAB	—
<a href="#">max</a>	MATLAB	—
<a href="#">max</a>	Fixed-Point Toolbox	—
<a href="#">mean</a>	MATLAB	—
<a href="#">median</a>	MATLAB	—
<a href="#">meshgrid</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not support character arrays</li> </ul>
<a href="#">min</a>	MATLAB	—
<a href="#">min</a>	Fixed-Point Toolbox	—
<a href="#">minus</a>	MATLAB	—
<a href="#">minus</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Any non-<code>fi</code> input must be constant; that value must be known at compile time so it can be cast to a <code>fi</code></li> </ul>
<a href="#">mldivide</a>	MATLAB	—
<a href="#">mod</a>	MATLAB	—
<a href="#">mpower</a>	MATLAB	—
<a href="#">mrdivide</a>	MATLAB	—

that is
al, but e complex passing
n different
is, its that it



<a href="#">mtimes</a>	MATLAB	—
<a href="#">mtimes</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>Any non-<code>fi</code> input must be constant; that value must be known at compile time so can be cast to a <code>fi</code></li> </ul>
NaN or <a href="#">nan</a>	MATLAB	<ul style="list-style-type: none"> <li>Dimensions must be real, non-negative, constants</li> <li>Supports only one or two dimension arguments</li> </ul>
<a href="#">nargin</a>	MATLAB	—
<a href="#">nargout</a>	MATLAB	—
<a href="#">ndims</a>	MATLAB	—
<a href="#">ndims</a>	Fixed-Point Toolbox	—
<a href="#">ne</a>	MATLAB	—
<a href="#">ne</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>Not supported for fixed-point signals with biases</li> </ul>
<a href="#">norm</a>	MATLAB	—
<a href="#">not</a>	MATLAB	—
<a href="#">numberofelements</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li><code>numberofelements</code> and <code>numel</code> both work the same as MATLAB <a href="#">numel</a> for <code>fi</code> objects Embedded MATLAB</li> </ul>
<a href="#">numerictype</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>Fixed-point signals coming in to an Embedded MATLAB Function block from Simulink are automatically assigned a <code>numerictype</code> object that is populated with the signal's data type and other information</li> <li>Returns the data type when the input is a nonfixed-point signal</li> </ul>
<a href="#">ones</a>	MATLAB	Dimensions must be real, non-negative, integer
<a href="#">or</a>	MATLAB	—
<a href="#">pascal</a>	MATLAB	—
<a href="#">pi</a>	MATLAB	—
<a href="#">pinv</a>	MATLAB	—
<a href="#">plus</a>	MATLAB	—
<a href="#">plus</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>Any non-<code>fi</code> input must be constant; that value must be known at compile time so can be cast to a <code>fi</code></li> </ul>
<a href="#">polyfit</a>	MATLAB	<ul style="list-style-type: none"> <li>Supports only one output.</li> </ul>

is, its that it
integer
uments
n different
work the in
edded re
d scaling
a
constants
is, its that it

<a href="#">polyval</a>	MATLAB	<ul style="list-style-type: none"> <li>• Supports only two input arguments and c argument.</li> </ul>
<a href="#">pow2</a>	Fixed-Point Toolbox	—
<a href="#">power</a>	MATLAB	<ul style="list-style-type: none"> <li>• Returns NaN when both X and Y are real. <code>power(X,Y)</code> is complex. To get the complex result, make the input value X complex by using <code>complex(X)</code>. For example, <code>power(complex(X),Y)</code>.</li> <li>• Returns NaN when both X and Y are real. <code>.^ Y</code> is complex. To get the complex result, make the input value X complex by using <code>complex(X)</code>. For example, <code>complex(X).^ Y</code>.</li> </ul>
<a href="#">prod</a>	MATLAB	—
<a href="#">qr</a>	MATLAB	—
<a href="#">range</a>	Fixed-Point Toolbox	—
<a href="#">rdivide</a>	MATLAB	—
<a href="#">real</a>	MATLAB	—
<a href="#">real</a>	Fixed-Point Toolbox	—
<a href="#">realmax</a>	MATLAB	—
<a href="#">realmax</a>	Fixed-Point Toolbox	—
<a href="#">realmin</a>	MATLAB	—
<a href="#">realmin</a>	Fixed-Point Toolbox	—
<a href="#">rem</a>	MATLAB	—
<a href="#">repmat</a>	MATLAB	—
<a href="#">repmat</a>	Fixed-Point Toolbox	—
<a href="#">rescale</a>	Fixed-Point Toolbox	—
<a href="#">reshape</a>	MATLAB	<ul style="list-style-type: none"> <li>• Accepts a maximum of three arguments</li> </ul>
<a href="#">reshape</a>	Fixed-Point Toolbox	<ul style="list-style-type: none"> <li>• Supported for 1-D and 2-D arrays only</li> </ul>
<a href="#">rot90</a>	MATLAB	—
<a href="#">round</a>	MATLAB	—

one output

, but  
complex  
by passing

$$X \rightarrow Y.$$
[illegible]

<a href="#">sec</a>	MATLAB	—
<a href="#">secd</a>	MATLAB	—
<a href="#">sech</a>	MATLAB	—
<a href="#">sign</a>	MATLAB	—
<a href="#">sign</a>	Fixed-Point Toolbox	—
<a href="#">sin</a>	MATLAB	—
<a href="#">sind</a>	MATLAB	—
<a href="#">single</a>	MATLAB	—
<a href="#">single</a>	Fixed-Point Toolbox	—
<a href="#">sinh</a>	MATLAB	—
<a href="#">size</a>	MATLAB	—
<a href="#">size</a>	Fixed-Point Toolbox	—
<a href="#">sort</a>	MATLAB	—
<a href="#">sosfilt</a>	Signal Processing Toolbox	—
<a href="#">sqrt</a>	MATLAB	<ul style="list-style-type: none"> <li>• Returns NaN when the input value <math>x</math> is real and the output should be complex. To get the complex result, make the input value complex by using <code>complex(x)</code>.</li> </ul>
<a href="#">strcmp</a>	MATLAB	—
<a href="#">sub2ind</a>	MATLAB	<ul style="list-style-type: none"> <li>• Does not support N-dimensional matrices. Input vector must have exactly two elements.</li> <li>• Maximum number of input arguments is 3.</li> </ul>
<a href="#">subsasgn</a>	Fixed-Point Toolbox	—
<a href="#">subsref</a>	Fixed-Point Toolbox	—
<a href="#">sum</a>	MATLAB	—
<a href="#">sum</a>	Fixed-Point Toolbox	—
<a href="#">svd</a>	MATLAB	—