HELP


Keywords are separated into commands and operators

Operators can also be special characters and sequences of multiple special characters


===

=  Operator help

===


| ! | Unary conventional factorial operator |
|---|---|
| # | Array indexing operation implemented as a binary operator |
| ## | Binomial coefficient operator (n ## k) |
| #/ | Factorial falling operator |
| #/# | Arithmetic division operator (expressed as fraction) |
| $# | Mark function call for Trapezoidal integral approximation |
| $% | Mark function call for adjustment for brute force approximation |
| $@ | Mark function call for Clenshaw-Curtis integral approximation |
| $\| | Mark function call for Tanh-Sinh integral approximation |
| % | Binary conventional remainder operator n%m; integer only |
| & | Logical AND |
| ' | Mark function call for first derivative approximation |
| '' | Mark function call for second derivative approximation |
| * | Arithmetic multiplication operator |
| ** | Binary conventional exponentiation operator x**y |
| *10^ | Decimal Shift |

*\        Conventional root operator a *\ b; a * sqrt(b)

*^#       Evaluate an exponential ( a * exp (b * x) ) defined by an array containing (a : b)

*^*       Compute tensor product of two matrices

+         Arithmetic addition operator

+#*       Evaluate a harmonic series defined by an array of coefficients

          ( c0 + c1*cos(x) + c2*cos(2x) + ... )

+*^       Evaluate a polynomial defined by an array of coefficients ( c0 + c1*x + c2*x^2 + ... )

+|-       Plus or Minus operator

-         Arithmetic subtraction operator

-#        Matrix indexing operation selecting row vector

-|+       Minus or Plus operator

.         Dot product of two arrays; lengths of arrays must match

..        Array range operation implemented as a binary operator

/         Arithmetic division operator

/#        Factorial rising operator

:         Choice based on condition code

<         Less than

<*>       Integral delta marker for presentation

| | |
|---|---|
| << | Left shift operator |
| <<= | Logical implied by |
| <= | Less than or equal to |
| <> | Derivative approximation evaluated at left parameter |
| | using delta value in right parameter |
| <\|\| | Less than absolute value of |
| == | Equal to |
| =>> | Logical implies |
| > | Greater than |
| >= | Greater than or equal to |
| >> | Right shift operator |
| >\|\| | Greater than absolute value of |
| ? | Logical condition code set |
| @# | Evaluate function defined by array at X |
| @*^ | Evaluate a Chebyshev polynomial using Clenshaw's special case |
| | defined by an array of coefficients ( $c0 + c1*T[1](x) + c2*T[2](x) + ...$ ) |

| | |
|---|---|
| ADJ | Compute adjugate of matrix |
| APPEND | Append a series of arrays into one long array |
| ARRAYDER | Compute derivative of function described by array |
| ARRAYINT | Compute integral of function described by array |
| AUGMENTED | Construct augmented matrix from source matrix and additional column |
| BERNOULLI | Bernoulli function B(m) for second (n=1) Bernoulli numbers |
| CHARACTERISTIC | Compute characteristic polynomial for matrix |
| CHEBINTERP | Generate Chebyshev interpolation polynomial for function described by array |
| CHEBYSHEV | Apply Vandermonde matrix to solve for Chebyshev interpolation polynomial |
| | as curve of best fit |
| CLENQUAD | Compute integral of function described by Chebyshev polynomial |
| COFACTOR | Compute cofactor matrix from source |
| COL | Read column vector from matrix |
| COMATRIX | Compute comatrix matrix from source |
| CONV | Compute product of polynomials |
| COV | Computed co-variance of an array of values |
| DECONV | Compute quotient of polynomials |
| DET | Compute determinant of matrix |
| DOT | Dot product of two arrays; lengths of arrays must match |
| DYADIC | Compute dyadic product of 2 arrays |
| EIG | Compute Von Mises dominant eigen-pair |
| EVALSPLINE | Evaluate a VC31 spline function at specified parameter |
| FALSE | Logical FALSE |

| | |
|---|---|
| FITEXP | Apply non-linear (logarithmic) regression to find curve of best fit |
| FITHARMONIC | Apply harmonic series regression to find curve of best fit |
| FITLINE | Apply least squares regression to find line of best fit |
| FITPOLY | Apply Vandermonde matrix to solve for polynomial coefficients to find curve of best fit |
| GAMMA | Gamma function |
| GAUSSIAN | Solve linear equations with Gaussian elimination |
| GAUSSQUAD | Apply Gauss Quadrature to build an integral of an interpolated Lagrange polynomial |
| GENKNOT | Construct a zero knot for odd or even functions |
| HARMONIC | Harmonic function H(x) |
| HYPOT | Distance in multi-dimensional space; SQRT of sum of squares of array elements |
| IDENTITY | Compute identity matrix with specified size |

| | |
|---|---|
| INTEGRAL | Sum of items of an array constructed based on delta terms; using traditional integral notation |
| INTEGRALC | Sum of items of an array constructed based on delta terms; using traditional contour integral notation |
| INTEGRALD | Sum of items of an array constructed based on delta terms; using traditional double integral notation |
| INTEGRALI | Sum of items of an array constructed based on delta terms; using traditional indefinite integral notation |
| INTEGRALS | Sum of items of an array constructed based on delta terms; using traditional surface integral notation |
| INTEGRALT | Sum of items of an array constructed based on delta terms; using traditional triple integral notation |
| INTEGRALV | Sum of items of an array constructed based on delta terms; using traditional volume integral notation |

| | |
|---|---|
| INTERPOLATE | Generate Lagrange interpolation polynomial for function described by array |
| INTERVAL | Select sub-list of elements for interval lo-hi |
| INV | Compute inverse of matrix |
| LAGRANGE | Apply Lagrange series to derive interpolation polynomial as curve of best fit |
| LENGTH | Length of an array treated as a unary function |
| LOGGAMMA | LogGamma function |
| LUXB | Solve LUx=b general case from assignment array=LUXB(L;U;b) |
| MATADD | Compute sum of two matrices |
| MATMUL | Compute product of two matrices |
| MATRIX | Construct matrix from array with dimensions |
| MAX | Maximum value found in array |
| MEAN | Mean of an array of values |
| MEDIAN | Median of an array of values |
| MIN | Minimum value found in array |
| MINOR | Compute minor of matrix |
| MINUS | Arithmetic subtraction operator |
| Mode | Mode of an array of values |
| NEGATE | Arithmetic negate operator |
| PEARSON | Compute Pearson regression coefficient for X/Y data set pair |
| PI | Product of items of an array; traditional capital PI notation |
| PIVOT | Reorder a vector to a specified pattern |
| POLYDER | Compute derivative of polynomial |
| POLYHG | Compute coefficients of hyper geometric polynomial |
| POLYINT | Compute integral of polynomial |

| | |
|---|---|
| \ | Binary conventional root operator n\\x; intended for small integer roots |
| \# | Matrix indexing operation selecting diag vector |
| ^ | Binary conventional exponentiation operator x^n; intended for small integer exponents |
| abs | Absolute value of parameter |
| asin | Trigonometric ARC SIN function |
| atan | Trigonometric ARC TAN function |
| cos | Trigonometric COS function |
| coscb | Trigonometric COS Cubed function |
| cossq | Trigonometric COS Squared function |
| e | Symbol for the irrational value of e; Epsilon base of natural log |
| epsilon | Epsilon base of natural log |
| exp | Unary conventional EXP function e^x |
| ln | Unary conventional natural logarithm function |
| pi | Symbol for the irrational value of pi; Trigonometric ratio of circle diameter to perimeter |
| sgn | Sign SGN function value of parameter |
| sin | Trigonometric SIN function |
| sincb | Trigonometric SIN Cubed function |
| sinsq | Trigonometric SIN Squared function |
| sqrt | Unary conventional SQRT function |
| tan | Trigonometric TAN function |
| tancb | Trigonometric TAN Cubed function |
| tansq | Trigonometric TAN Squared function |
| zeta | Zeta function |

| Symbol | Description |
|--------|-------------|
| \| | Logical OR |
| \|# | Matrix indexing operation selecting column vector |
| \|\| | Mark function call for interval evaluation |
| \|~ | Logical XOR |
| ~ | Logical NOT |
| ~& | Logical NAND |
| ~<<= | Logical NOT implied by |
| ~= | Not equal to |
| ~=>> | Logical NOT implies |
| ~\| | Logical NOR |
| ~\|~ | Logical NOT XOR |

!!        Define a user function

!$        Define a segmented user function as a list of polynomials

!%        Define a function transform

!*        Declare a user function as a Hyper-Geometric polynomial

!+        Define a user function as a library import

!^        Define a function as an integration transform source

//        Add a comment to the output stream

ASSERT          Make a conditional assertion

BACKGROUND  Read a script file and execute as a background task

CALC            Calculate and show value for an expression

CALCULATE       Calculate and show value for an expression

DCT             Apply Discrete Cosine Tranform to function

DEFINE          Define a user function

DERIVE          Plot derivatives of a polynomial

ENCODE          Encode a segmented function as a Java class

ENTITLED        Change the title of the last frame displayed

EXPORT          Export data to a file from specified matrix

FAMILY          Import polynomial power functions for named family

| | |
|---|---|
| FFT | Compute and display a fast Fourier transform |
| GRAPH | Display a graph of an array of data points |
| HELP | Show the HELP table |
| IMPORT | Import data from a file into specified matrix |
| ITERATE | Read a script file and iterate |
| LIBRARY | Construct a library of functions |
| MANDELBROT | Display a plot of the Mandelbrot set |
| PLOT3D | Plot a specified 2D domain of a 3D user defined function |
| PLOTF | Plot a specified range of a user defined function |
| POLY | Analyze a polynomial and tabulate key points |
| POLYNOMIAL | Analyze a polynomial and tabulate key points |
| POLYPRINT | Format an array as a polynomial |
| PRETTYPRINT | Show the formatted value of a symbol |
| PRIMEGAPS | Tabulate gaps between primes |
| PRIMETABLE | Tabulate primes starting from specified |
| RADIX | Calculate and show value for an expression in specified radix |
| READ | Read a workspace storage file |
| RECOGNIZE | Read a symbol definition file |
| RENDER | format and display (pretty print) an equation using MathML |
| RENDERF | format and display (pretty print) a function using MathML |
| RPN | Start the RPN calculator |
| RUNSIEVE | Run a sieve to populate the primes table |
| SAVE | Save a workspace storage file |
| SCATTER | Produce an X/Y scatter plot |
| SCRIPTPRINT | Display the contents of a script file |

SETCONTOUR    Set color scheme manager for contour plots

SETDOMAIN     Set domain constraints on a function

SETMODE       Set value display mode and precision

SHOW          Show symbol table contents Symbols|Functions|Parent|ALL parents

SPLOT         Display a graph of a complex number sequence

STDDOMAIN     Standardize a function domain to [-1,1]

VERIFY        Verify symbol present in current symbol table