Mathcad

PTC

Mathcad Mathematical Functions

Mathcad and the Mathcad Extension Packs give you access to hundreds of powerful mathematical functions that enable you to solve your critical scientific and engineering calculations with ease.

Mathcad Functions

Function Name (Parameters)	Function Definition
acos (z)	Returns the angle (in radians) whose cosine is z. Principal value for complex z.
acosh (z)	Returns the angle (in radians) whose hyperbolic cosine is z. The result is the principal value for complex z.
acot (z)	Returns the angle (in radians) whose cotangent is z. The result is between 0 and pi if z is real, and the principal value if z is complex.
acoth (z)	Returns the angle (in radians) whose hyperbolic cotangent is z. The result is the principal value for complex z.
acsc (z)	Returns the angle (in radians) whose cosecant is z. The result is the principal value for complex z.
acsch (z)	Returns the angle (in radians) whose hyperbolic arccosecant is z. The result is the principal value for complex z.
Adams (y, x1, x2, npoints, D, [tol])	Returns a matrix of solution values for the differential equation specified by the derivatives in D, and having initial conditions y on the interval [x1,x2], using Adams methods. Parameter npoints controls the number of rows in the matrix output.
AdamsBDF (y, x1, x2, npoints, D, [J], [tol])	Returns a matrix of solution values for the differential equation specified by the derivatives in D, and having initial conditions y on the interval [x1,x2], using BDF methods for stiff systems and Adams methods for non-stiff systems. Parameter npoints controls the number of rows in the matrix output.
Ai (z)	Returns the value of the Airy function of the first kind.
Ai.sc (x)	Returns the value of the Airy function of the first kind, scaled by the factor Re(exp(2/3*z^(3/2))).
angle (x, y)	Returns the angle (in radians) between the x-axis and the point (x, y). x and y must be real.
antisymmetric tensor (i, j, k)	Returns the completely antisymmetric tensor of rank 3. Result is 0 if any two arguments are the same, 1 for even permutations, -1 for odd permutations..
APPENDPRN (file, [M])	Writes the contents of an array to the end of a delimited ASCII file.
arg (z)	Returns the principal argument of the complex number z, between -pi and pi, including pi.

asec (z)	Returns the angle (in radians) whose secant is z. The result is the principal value for complex z.
asech (z)	Returns the angle (in radians) whose hyperbolic secant is z. Result is the principal value for complex z.
asin (z)	Returns the angle (in radians) whose sine is z. Principal value for complex z.
asinh (z)	Returns the angle (in radians) whose hyperbolic sine is z. Principal value for complex z.
atan (z)	Returns the angle (in radians) whose tangent is z. Principal value for complex z.
atan2 (x, y)	Returns the angle (in radians) from the x-axis to a line containing the origin (0,0) and the point (x,y). Both x and y must be real.
atanh (z)	Returns the angle (in radians) whose hyperbolic tangent is z. Principal value for complex z.
augment (A, B, C, ...)	Returns an array formed by placing A, B, C, ... left to right
BDF (y, x1, x2, npoints, D, [J], [tol])	Returns a matrix of solution values for the stiff differential equation specified by the derivatives in D, and initial conditions y on the interval [x1,x2], using backwards differentiation formula methods. Parameter npoints controls the number of rows in the matrix output.
bei (m, x)	Returns the value of the imaginary Bessel Kelvin function of order m.
ber (m, x)	Returns the value of the real Bessel Kelvin function of order m.
Bi (x)	Returns the value of the Airy function of the second kind.
Bi.sc (x)	Returns the value of the Airy function of the second kind, scaled by the factor exp(-|Re(2/3*z^3/2)|).
bspline (vx, vy, u, n)	Returns a vector of the coefficients of a B-spline of degree n for the data in vx and vy, given the knot values in u. The vector returned becomes the first argument of the interp function.
Bulstoer (y, x1, x2, npoints, D)	Returns a matrix of solution values for the smooth differential equation specified by the derivatives in D using a Bulirsch-Stoer method.
bvalfit (v1, v2, x1, x2, xf, D, load1, load2, score)	Returns a vector of initial conditions for the boundary value problem specified by the derivatives in D where the solution is known at the intermediate point xf.
ceil (z)	Returns the smallest integer greater than or equal to z.
Ceil (z, y)	Returns the smallest multiple of y greater than or equal to z, typically used for correct unit scaling.
cfft (A)	Returns the Discrete Fourier transform of any size vector or matrix of real or complex numbers. Returns an array of the same size as its argument.
CFFT (A)	Returns the Discrete Fourier transform of any size vector or matrix of real or complex numbers. Returns an array of the same size as its argument. Similar to cfft(A), with a different normalizing factor and sign convention.
Chi (x)	Returns the value of the hyperbolic cosine integral function. Can only be evaluated symbolically.

cholesky (M)	Returns the lower triangular matrix L such that L times L transpose is M. L is the cholesky square root of the input matrix.
Ci (x)	Returns the value of the cosine integral function. Can only be evaluated symbolically.
cnorm (x)	Returns the cumulative probability distribution with mean 0 and variance 1.
cnper (rate, pv, fv)	Returns the number of compounding periods for an investment to yield a specified future value.
cols (A)	Returns the number of columns in A.
combin (n, k)	Returns the number of subsets (combinations) of k elements that can be formed from n elements.
concat (S1, S2, S3, ...)	Returns the string formed by concatenating strings S1, S2, and so on.
cond1 (M)	Returns the condition number of the matrix M based on the L1 norm.
cond2 (M)	Returns the condition number of the matrix M based on the L2 norm.
conde (M)	Returns the condition number of the matrix M based on the Euclidean norm.
condi (M)	Returns the condition number of the matrix M based on the infinity norm.
corr (A, B)	Returns the Pearson's r correlation coefficient of the elements in A and B.
correl (vx, vy)	Returns the correlation of vectors vx and vy. Result is a vector for which each element contains the summed vector product of vx and a shifted version of vy.
correl2d (M, K)	Returns the 2D correlation of matrix M with kernel K. The resulting matrix contains the summed element-wise product of K overlapped with a subset of M.
cos (z)	Returns the cosine of z.
cosh (z)	Returns the hyperbolic cosine of z.
cot (z)	Returns the cotangent of z.
coth (z)	Returns the hyperbolic cotangent of z.
crate (nper, pv, fv)	Returns the fixed interest rate per period required for an investment at present value to yield a specified future value over a number of compounding periods.
CreateMesh (function(s), [s0, s1, t0, t1], [sgrid, tgrid], [fmap])	Returns a nested array of three matrices representing the x-, y-, and z-coordinates of a parametric surface defined by the function(s) of two variables in the first argument(s).
CreateSpace (function(s), [t0, t1], [tgrid], [fmap])	Returns a nested array of three vectors representing the x-, y-, and z-coordinates of a space curve defined by the function(s) of one variable in the first argument.
csc (z)	Returns the cosecant of z.
csch (z)	Returns the hyperbolic cosecant of z.
csgn (z)	Returns the complex sign of z, given by 0 if z = 0, 1 if the real or imaginary part of z is > 0, and -1 otherwise.
csort (A, n)	Returns an array formed by rearranging rows of A until column n is in ascending order.

cspline (vx, vy)	Returns a vector of cubic spline coefficients with cubic endpoints which fits the independent data in vector or matrix vx and dependent data in vy. This vector becomes the first argument of the interp function.
cumint (rate, nper, pv, start, end, [type])	Returns the cumulative interest paid on a loan between a starting period and an ending period given a fixed interest rate, the total number of compounding periods, and the present value of the loan.
cumprn (rate, nper, pv, start, end, [type])	Returns the cumulative principal paid on a loan between a starting period and an ending period given a fixed interest rate, the total number of compounding periods, and the present value of the loan.
cvar (A, B)	Returns the covariance of the elements in A and B.
cyl2xyz (r, q, f)	Converts the cylindrical coordinates of a point in 3D space to rectangular coordinates.
DAi (z)	Returns the value of the first derivative of the Airy function of the first kind.
DAi.sc (x)	Returns the value of the first derivative of the Airy function of the first kind, scaled by the factor Re(exp(2/3*z^(3/2))).
dbeta (x, s1, s2)	Returns the probability density for the beta distribution with shape parameters s1 and s2.
DBi (z)	Returns the value of the first derivative of the Airy function of the second kind.
DBi.sc (x)	Returns the value of the first derivative of the Airy function of the second kind, scaled by the factor exp(-|Re(2/3*z^3/2)|).
dbinom (k, n, q)	Returns the probability density for the Binomial distribution.
dcauchy (x, l, s)	Returns the probability density for the Cauchy distribution with location l and scale s.
dchisq (x, d)	Returns the probability density for the chi-squared distribution with degrees of freedom d.
denom (x)	Returns the denominator of the fraction or rational expression x. Can only be evaluated symbolically.
dexp (x, r)	Returns the probability density for the exponential distribution with rate of decay r.
dF (x, d1, d2)	Returns the probability density for the F distribution with degrees of freedom d1 and d2.
dgamma (x, s)	Returns the probability density for the gamma distribution with shape parameter s.
dgeom (k, p)	Returns the probability density for the Geometric distribtuion, with probability of success p.
dhypergeom (m, a, b, n)	Returns the probability density for the hypergeometric distribution.
diag (v)	Returns a matrix containing on its diagonal the elements of v.
dilog (x)	Returns the value of the dilogarithm function. Can only be evaluated symbolically.
dlnorm (x, mu, sigma)	Returns the probability density for the lognormal distribution with logmean mu and logdeviation sigma.
dlogis (x, l, s)	Returns the probability density for the logistic distribution with location l and scale s.

DMS (x)	Returns the angle in radians given a vector containing degrees, minutes, and seconds; or returns the vector given the angle when used in the units placeholder.
dnbinom (k, n, p)	Returns the probability density for the negative binomial distribution with size n and probability of failure p.
dnorm (x, mu, sigma)	Returns the probability density for the normal distribution with mean mu and standard deviation sigma.
dpois (k, l)	Returns the probability density for the Poisson distribution in which mean l.
dt (x, d)	Returns the probability density for Student's t distribution with degrees of freedom d.
dunif (x, a, b)	Returns the probability density for the uniform distribution on an interval [a,b].
dweibull (x, s)	Returns the probability density for the Weibull distribution with shape parameter s.
eff (rate, npery)	Returns the effective annual interest rate (APR), given the nominal interest rate and the number of compounding periods per year.
Ei (x)	Returns the value of the exponential integral function.
eigenvals (M)	Returns a vector of eigenvalues for the square matrix M.
eigenvec (M, z)	Returns the normalized eigenvector associated with eigenvalue z of the square matrix M. The eigenvector is normalized to unit length.
eigenvecs (M)	Returns a matrix containing the normalized eigenvectors corresponding to the eigenvalues of the square matrix M. The nth column of the matrix returned is an eigenvector corresponding to the nth eigenvalue returned by eigenvals. Optional last argument "L" specifies the left eigenvalue.
erf (z)	Returns the error function.
erfc (x)	Returns the complementary error function.
error (S)	Returns the string S as a Mathcad error tip.
exp (z)	Returns the number e raised to the power z.
expfit (vx, vy, [vg])	Returns a vector containing three coefficients for an exponential curve of the form a*e^(b*x)+c that best approximates the data in vectors vx and vy. The optional vector vg contains guess values for the three coefficients.
fft (v)	Returns the fast Fourier transform of real data vector v with 2^n elements. Returns a vector of size 2^n-1 + 1. Similar to FFT(v), except uses a different normalizing factor and sign convention.
FFT (v)	Returns the fast Fourier transform of real data vector v with 2^n elements. Returns a vector of size 2^n-1 + 1. Similar to fft(v), except uses a different normalizing factor and sign convention.
fhyper (a, b, c, x)	Returns the value of the Gauss hypergeometric function at the point x given parameters a, b, c.
FIF (x)	Returns a length given a string representing feet-inches-fractions; or returns the FIF string given a length when used in the units placeholder.

Find (var1, var2, ...)	Returns the values of var1, var2, ..., that solve a system of equations in a Solve Block. Returns a scalar if there is only one argument, otherwise returns a vector of answers.
floor (z)	Returns the greatest integer less than or equal to z.
Floor (z, y)	Returns the greatest multiple of y less than or equal to z, typically used for correct unit scaling.
format (S, x, y, z, ...)	Returns a string containing the value of the arguments x, y, z, ... with print order and surrounding text specified by S. S is optional if only one value is printed.
FresnelC (x)	Returns the value of the Fresnel cosine integral function.
FresnelS (x)	Returns the value of the Fresnel sine integral function.
fv (rate, nper, pmt, [[pv], [type]])	Returns the future value of an investment or loan given a periodic, constant payment and a fixed interest rate.
fvadj (prin, v)	Returns the future value of an initial principal after applying the series of compound interest rates in vector v.
fvc (rate, v)	Returns the future value of a vector of cash flows, v, earning a specified interest rate.
Gamma ([a], z)	Returns either Euler's gamma function of z, or the incomplete gamma function of z with degree a. To type G, press G+[Ctrl]+G.
gcd (A, B, C, ...)	Returns the greatest common divisor: the largest integer that evenly divides all the elements of A, B, C, ...
genfit (vx, vy, vg, F)	Returns a vector of parameters that make the first function in the vector F best fit the data in the vectors vx and vy. The remaining elements in F are partial derivatives of the fitting function with respect to its n parameters, and vg is a vector of guess values. Right-click on this function to choose a solver.
geninv (A)	Returns the generalized (pseudo) inverse of the input matrix A, giving the least-squares solution to a system of equations..
genvals (M, N)	Returns a vector of eigenvalues which satisfy the generalized eigenvalue problem.
genvecs (M, N)	Returns a matrix of normalized eigenvectors corresponding to the eigenvalues returned by genvals. Optional last argument "L" specifies the left eigenvalue.
GETWAVINFO (file)	Returns a vector containing, in order, the number of channels, the sample rate, the bit resolution, and the average bytes per second for a WAV file.
gmean (A, B, C, ...)	Returns the geometric mean of the elements of A, B, C, ...
H1 (m, z)	Returns the Hankel function of the first kind (Bessel function of the third kind).
H1.sc (m, z)	Returns the Hankel function of the first kind (Bessel function of the third kind), scaled by the factor exp(-z*i).
H2 (m, z)	Returns the Hankel function of the second kind (Bessel function of the third kind).
H2.sc (m, z)	Returns the Hankel function of the second kind (Bessel function of the third kind), scaled by the factor exp(z*i).
heaviside step (x)	Returns the heaviside step function with value 1 if x is greater than or equal to 0, 0 otherwise. To type Phi, press F+[Ctrl]+G.

Her (n, x)	Returns the value of the Hermite polynomial of degree n at x.
hhmmss (x)	Returns a time given a string containing hours:minutes:seconds; or returns the string given a time when used in the units placeholder.
hist (intvls, data)	Returns a vector representing the frequencies with which values in data fall into the intervals represented by intvls. intvls can be a vector of interval endpoints, or an integer number of subintervals of equal length.
histogram (intvls, data)	Returns a two-column matrix containing the midpoints of the intvls subintervals. The second column is identical to the vector returned by hist. The resulting matrix has intvls rows.
hlookup (z, A, r)	Looks in the first row of a matrix, A, for a given value, z, and returns the value(s) in the same column(s) in the row specified, r. When multiple values are returned, they appear in a vector.
hmean (A, B, C, ...)	Returns the harmonic mean of the elements of A, B, C, ...
I0 (z)	Returns the zeroth order modified Bessel function of the first kind.
I0.sc (z)	Returns the zeroth order modified Bessel function of the first kind, scaled by the factor exp(-|Re(z)|).
I1 (z)	Returns the first order modified Bessel function of the first kind.
I1.sc (z)	Returns the first order modified Bessel function of the first kind, scaled by the factor exp(-|Re(z)|).
ibeta (a, x, y)	Returns the value of the incomplete beta function of x and y with parameter a.
icfft (A)	Returns the inverse Fourier transform corresponding to cfft. Returns an array of the same size as its argument.
ICFFT (A)	Returns the inverse Fourier transform corresponding to CFFT. Returns an array of the same size as its argument.
identity (n)	Returns an n x n identity matrix (a matrix of 0's with 1's along the diagonal).
if (cond, x, y)	Returns x if logical condition cond is true (non-zero), y otherwise.
ifft (u)	Returns the inverse Fourier transform corresponding to fft. Takes a vector of size 1 + 2^n-1, where n is an integer. Returns a real vector of size 2^n.
IFFT (u)	Returns the inverse Fourier transform corresponding to FFT. Takes a vector of size 1 + 2^n-1, where n is an integer. Returns a real vector of size 2^n.
Im (z)	Returns the imaginary part of complex number, vector, or matrix, z.
In (m, z)	Returns the mth order modified Bessel function of the first kind.
In.sc (m, z)	Returns the mth order modified Bessel function of the first kind, scaled by the factor exp(-|Re(z)|).
intercept (vx, vy)	Returns the intercept of line that best fits data in vx and vy.

interp (vs, vx, vy, x)	Returns an interpolated value at x from the coefficients in vector vs, and the original data in vx and vy. Coefficient vector vs is the output of one of the following: cspline, lspline, pspline, bspline, loess, or regress.
ipmt (rate, per, nper, pv, [[fv], [type]])	Returns the interest payment of an investment or loan for a given period based on periodic, constant payments over a given number of compounding periods using a fixed interest rate and a specified present value.
irr (v, [guess])	Returns the internal rate of return for a series of cash flows occurring at regular intervals.
IsArray (x)	Returns 1 if x is a matrix or vector. Returns 0 otherwise.
IsNaN (x)	Returns 1 if x is NaN. Returns 0 otherwise.
IsPrime (n)	Returns 1 if n is prime and 0 otherwise. Can only be evaluated symbolically.
IsScalar (x)	Returns 1 if x is a real or complex scalar. Returns 0 otherwise.
IsString (x)	Returns 1 if x is a string. Returns 0 otherwise.
iwave (v)	Returns the inverse one-dimensional discrete wavelet transform of v computed using the Daubechies 4-coefficient wavelet filter in the wave function.
J0 (z)	Returns the zeroth order Bessel function of the first kind.
J0.sc (z)	Returns the zeroth order Bessel function of the first kind, scaled by exp(-|Im(z)|).
J1 (z)	Returns the first order Bessel function of the first kind.
J1.sc (z)	Returns the first order Bessel function of the first kind, scaled by exp(-|Im(z)|).
Jac (n, a, b, x)	Returns the value of the Jacobi polynomial of degree n at x with parameters a and b.
Jacob (F(x),x,[k])	Returns the Jacobian matrix of the vector function F(x).
Jn (m, z)	Returns the mth order Bessel function of the first kind.
Jn.sc (m, z)	Returns the mth order Bessel function of the first kind, scaled by exp(-|Im(z)|).
js (m, z)	Returns the value of the spherical Bessel function of the first kind, of order m.
K0 (z)	Returns the zeroth order modified Bessel function of the second kind.
K0.sc (z)	Returns the zeroth order modified Bessel function of the second kind, scaled by the factor exp(z).
K1 (z)	Returns the first order modified Bessel function of the second kind.
K1.sc (z)	Returns the first order modified Bessel function of the second kind, scaled by the factor exp(z).
Kn (m, z)	Returns the mth order modified Bessel function of the second kind.
Kn.sc (m, z)	Returns the mth order modified Bessel function of the second kind, scaled by the factor exp(z).
Kronecker delta (x, y)	Returns the Kronecker delta function with value 1 if x = y, 0 otherwise. To type Delta, press d+[Ctrl]+G.
ksmooth (vx, vy, b)	Returns a vector of local weighted averages of vy using a Gaussian kernel with bandwidth b.
kurt (A, B, C, ...)	Returns the kurtosis of the elements of A, B, C, ...

Lag (n, x)	Returns the value of the Laguerre polynomial of degree n at x.
LambertW ([n],x)	Lambert(x) returns the value of the Lambert W function. Lambert(n,x) returns the value of the nth branch of the Lambert W function.
last (v)	Returns the scalar index of the last element in vector v.
lcm (A, B, C, ...)	Returns the least common multiple: the smallest positive integer that is a multiple of all the elements of A, B, C, ...
Leg (n, x)	Returns the value of the Legendre polynomial of degree n at x.
length (v)	Returns the integer number of elements in vector v.
lgsfit (vx, vy, vg)	Returns a vector containing the 3 coefficients for a logistic curve of the form a/(1+b*e^(-c*x)) that best approximates the data in vectors vx and vy, using guess values in vg.
line (vx, vy)	Returns a vector containing the coefficients for a line of the form a + bx that best approximates the data in vectors vx and vy.
linfit (vx, vy, F)	Returns a vector containing the coefficients used to create a linear combination of the functions in vector F which best approximates the data in vx and vy.
linterp (vx, vy, x)	Returns a linearly interpolated value at x for data vectors vx and vy of the same size.
ln (z)	Returns the natural logarithm (base e) of z. Returns principal value (imaginary part between pi and -pi) for complex z.
ln0 (z)	Returns the natural logarithm (base e) of z but allows z = 0. Returns principal value (imaginary part between pi and -pi) for complex z.
lnfit (vx, vy)	Returns a vector containing the 2 coefficients for a logarithmic curve of the form a*ln(x) + b that best approximates the data in vx and vy.
lnGamma (z)	Returns the natural logarithm of Euler's gamma function, evaluated at z. To type G, press G[Ctrl]G.
LoadColormap (file)	Returns an array containing the values in the colormap named file.
loess (vx, vy, span)	Returns a vector used by the interp function to find a set of second order polynomials that best fit a neighborhood of data values in vectors or matrices vx and vy. The size of the neighborhood is controlled by span.
log (z, [b])	Returns the base b logarithm of z. If b is omitted, returns the base 10 logarithm.
logfit (vx, vy, vg)	Returns a vector containing the three coefficients for a logarithmic curve of the form a*ln(x+b)+c that best approximates the data in vectors vx and vy. Vector vg contains guess values for the three coefficients.
logpts (minexp, dec, dnpts)	Returns a vector with dec decades of evenly-spaced points starting at 10 raised to the exponent minexp, with dnpts points per decade.
logspace (min, max, npts)	Returns a vector of npts logarithmically-spaced points starting at min, ending at max.

lookup (z, A, B)	Looks in a vector or matrix, A, for a given value, z, and returns the value(s) in the same position(s) (i.e., with the same row and column numbers) in another matrix, B. When multiple values are returned, they appear in a vector.
lsolve (M, v)	Returns the vector x solving the linear system of equations M*x = v.
lspline (vx, vy)	Returns a vector of cubic spline coefficients with linear endpoints which fits the independent data in vector or matrix vx and dependent data in vy. This vector becomes the first argument of the interp function.
lu (M)	Returns a matrix containing three augmented square matrices P, L, and U, all having the same size as M; these satisfy the equation P M = L U.
match (z, A)	Looks in a vector or matrix, A, for a given value, z, and returns the index (indices) of its positions in A.
matrix (m, n, f)	Returns a m x n matrix in which the ijth element is given by f(i,j).
max (A, B, C, ...)	Returns the largest value from A, B, C, ... If any value is complex, returns max(Re(A, B, C, ...)) + i*max(Im(A, B, C, ...)).
Maximize (f, var1, var2, ...)	Returns the values of var1, var2, ..., that satisfy the constraints in a Solve Block, and make the function f take on its greatest value. Returns a scalar if there is only one argument, otherwise returns a vector of answers.
mean (A, B, C, ...)	Returns the arithmetic mean, or average, of the elements of A, B, C, ...
medfit (vx, vy)	Returns a vector containing the coefficients for a line of the form a + bx that best approximates the data in vectors vx and vy using median-median regression.
median (A, B, C, ...)	Returns the median of the elements of A, B, C ...
medsmooth (vy, n)	Returns a smoothed vector by replacing each value in vy with the median of the n points centered on that value.
mhyper (a, b, x)	Returns the value of the confluent hypergeometric function, M(a,b,x), at the point x with parameters a and b.
min (A, B, C, ...)	Returns the smallest value in A, B, C, ... If any value is complex, returns min(Re(A, B, C, ...)) + i*min(Im(A, B, C, ...)).
Minerr (var1, var2, ...)	Returns the values of var1, var2, ..., coming closest to satisfying a system of equations and constraints in a Solve Block. Returns a scalar if only one argument, otherwise returns a vector of answers. If Minerr cannot converge, it returns the results of the last iteration.
Minimize (f, var1, var2, ...)	Returns the values of var1, var2, ..., that satisfy the constraints in a Solve Block, and make the function f take on its smallest value. Returns a scalar if there is only one variable, otherwise returns a vector of answers.
mirr (v, fin_rate, rein_rate)	Returns the modified internal rate of return for a vector v of cash flows given a finance rate and a reinvestment rate.
mod (x, y)	Returns the remainder on dividing x by y (x modulo y). Result has the same sign as x.
mode (A, B, C, ...)	Returns the value in A, B, C, ... that occurs most often.

multigrid (M, ncycle)	Returns a square matrix of solution values for Poisson's partial differential equation, controlled by ncycle, in the case of zero boundaries. Otherwise, use the relax function.
nom (APR, npery)	Returns the nominal interest rate, given the effective annual interest rate (APR) and the number of compounding periods per year.
norm1 (M)	Returns the L1 norm of the matrix M.
norm2 (M)	Returns the L2 norm of the matrix M.
norme (M)	Returns the Euclidean norm of the matrix M.
normi (M)	Returns the infinity norm of the matrix M.
nper (rate, pmt, pv, [[fv], [type]])	Returns the number of compounding periods for an investment or loan based on periodic, constant payments using a fixed interest rate and a specified present value.
npv (rate, v)	Returns the net present value of an investment given a discount rate and a series of cash flows occuring at regular intervals.
num2str (z)	Returns the number z to a string.
numer (x)	Returns the numerator of the fraction or rational expression x. Can only be evaluated symbolically.
numol (x_endpts, xpts, t_endpts, tpts, num_pde, num_pae, pde_func, pinit, bc_func)	Returns an xpts by tpts matrix containing the solutions to the one-dimensional PDEs in pde_func. Each column represents a solution over 1-D space at a single solution time. For a system of equations, the solution for each function is appended horizontally, so the matrix always has xpts rows, and tpts * (num_pde + num_pae) columns.
Odesolve ([vf], x, b, [step])	Returns a function or vector of functions of x representing the solution to a system of ordinary differential equations in a Solve Block. vf is omitted when solving a single ODE.
pause (S, x, y, z, ...)	Returns a string containing the value of the arguments x, y, z, ... with print order and surrounding text specified by S. Prints values in the Trace Window and pauses execution when debug mode is on. S is optional if only one value is printed.
pbeta (x, s1, s2)	Returns the cumulative probability beta distribution with shape parameters s1 and s2.
pbinom (k, n, q)	Returns the cumulative probability binomial distribution for k successes in n trials, given a probability of success, q, per trial.
pcauchy (x, l, s)	Returns the cumulative probability Cauchy distribution with location l and scale s.
pchisq (x, d)	Returns the cumulative probability chi-squared distribution with degrees of freedom d.
Pdesolve (u, x, xrange, t, trange, [xpts], [tpts])	Returns a vector u of functions of x and t representing the solution to a system of partial differential equations in a Solve Block over the ranges specified in xrange and trange.
permut (n, k)	Returns the number of ways of ordering n distinct objects taken k at a time (permutations).
pexp (x, r)	Returns the cumulative exponential probability distribution with rate r.
pF (x, d1, d2)	Returns the cumulative F probability distribution with degrees of freedom d1 and d2.

pgamma (x, s)	Returns the cumulative gamma probability distribution with shape parameter s.
pgeom (k, p)	Returns the cumulative geometric probability distribution with probability of success p.
phypergeom (m, a, b, n)	Returns the cumulative hypergeometric probability distribution.
plnorm (x, mu, sigma)	Returns the cumulative lognormal probability distribution with logmean mu and logdeviation sigma.
plogis (x, l, s)	Returns the cumulative logistic probability distribution with location l and scale s.
pmt (rate, nper, pv, [[fv], [type]])	Returns the payment for an investment or loan based on periodic, constant payments over a given number of compounding periods using a fixed interest rate and a specified present value.
pnbinom (k, n, p)	Returns the cumulative negative binomial probability distribution with size n and probability of failure p.
pnorm (x, mu, sigma)	Returns the cumulative normal probability distribution with mean mu and standard deviation sigma.
pol2xy (r, theta)	Converts the polar coordinates of a point in 2D space to rectangular coordinates.
Polyhedron (S)	Generates the uniform polyhedron whose name, number code, or Wythoff symbol is string S.
PolyLookup (n)	Returns a vector containing the name, the dual name, and the Wythoff symbol for the polyhedron whose number code is n.
polyroots (v)	Returns a vector containing all the roots of the polynomial whose coefficients are in v. Right-click on this function to choose a solver.
ppmt (rate, per, nper, pv, [[fv], [type]])	Returns the payment on the principal of an investment or loan for a given period based on periodic, constant payments over a given number of compounding periods using a fixed interest rate and a specified present value.
ppois (k, l)	Returns the cumulative Poisson probability distribution in which l > 0.
predict (v, m, n)	Returns a vector of n predicted values past the last element in v, based on autocorrelation coefficients of m consecutive values in a sliding window.
Psi ([n],x)	Psi(x) returns the digamma function of x. Psi(n,x) returns the polygamma function of x. Can only be evaluated symbolically.
pspline (vx, vy)	Returns a vector of cubic spline coefficients with parabolic endpoints that fits the independent data in vector or matrix vx and dependent data in vy. This vector becomes the first argument of the interp function.
pt (x, d)	Returns the cumulative probability Student's t distribution with degrees of freedom d.
punif (x, a, b)	Returns the cumulative uniform probability distribution on the interval [a,b].

pv (rate, nper, pmt, [[fv], [type]])	Returns the present value of an investment or loan based on periodic, constant payments over a given number of compounding periods using a fixed interest rate and a specified payment.
pweibull (x, s)	Returns the cumulative Weibull probability distribution with shape parameter s.
pwrfit (vx, vy, vg)	Returns a vector containing the coefficients for a power curve of the form a*x^b + c that best approximates the data in vectors vx and vy. Vector vg contains guess values for the three coefficients.
qbeta (p, s1, s2)	Returns the inverse cumulative beta distribution with shape parameters s1 and s2.
qbinom (p, n, q)	Returns the inverse cumulative binomial distribution with size n and probability of success q.
qcauchy (p, l, s)	Returns the inverse cumulative Cauchy distribution with location l and scale s.
qchisq (p, d)	Returns the inverse cumulative chi-squared distribution with degrees of freedom d.
qexp (p, r)	Returns the inverse cumulative exponential distribution with rate r.
qF (p, d1, d2)	Returns the inverse cumulative F distribution with degrees of freedom d1 and d2.
qgamma (p, s)	Returns the inverse cumulative gamma distribution with shape parameter s.
qgeom (p, q)	Returns the inverse cumulative geometric distribution with probability of success q.
qhypergeom (p, a, b, n)	Returns the inverse cumulative probability distribution for the hypergeometric distribution. p is a real number between 0 and 1. a, b, and n are integers.
qlnorm (p, mu, sigma)	Returns the inverse cumulative lognormal distribution with logmean mu and logdeviation sigma.
qlogis (p, l, s)	Returns the inverse cumulative logistic distribution with location l and scale s.
qnbinom (p, n, q)	Returns the inverse cumulative negative binomial distribution with size n and probability of failure q.
qnorm (p, mu, sigma)	Returns the inverse cumulative normal distribution with mean mu and standard deviation sigma.
qpois (p, l)	Returns the inverse cumulative Poisson distribution with l > 0.
qr (A)	Returns a matrix whose first n columns contain the square, orthonormal matrix, Q, and whose remaining columns contain the upper triangular matrix, R, forming the QR decomposition of the input matrix Q*R = A.
qt (p, d)	Returns the inverse cumulative Student's t distribution with degrees of freedom d.
qunif (p, a, b)	Returns the inverse cumulative uniform distribution on the interval [a,b].
qweibull (p, s)	Returns the inverse cumulative Weibull distribution with shape parameter s.

Radau (y, x1, x2, npoints, D, [J], [M], [tol])	Returns a matrix of solution values for the stiff differential equation specified by the derivatives in D, and initial conditions y on the interval [x1,x2], using a RADAU5 method. Parameter npoints controls the number of rows in the matrix output.
rank (A)	Returns the rank of matrix A, the number of linearly independent columns.
rate (nper, pmt, pv, [[fv], [type], [guess]])	Returns the interest rate per period of an investment or loan over a specified number of compounding periods given a periodic, constant payment and a specified present value.
rbeta (m, s1, s2)	Returns a vector of m random numbers having the beta distribution with shape parameters s1 and s2.
rbinom (m, n, q)	Returns a vector of m random numbers having the binomial distribution with size n and probability of success q.
rcauchy (m, l, s)	Returns a vector of m random numbers having the Cauchy distribution with location l and scale s.
rchisq (m, d)	Returns a vector of m random numbers having the chi-squared distribution with degrees of freedom d.
Re (z)	Returns the real part of complex number z.
READ_BLUE (file)	Returns a matrix representing the RGB blue component of the BMP, GIF, JPG, or TGA color image in file.
READ_GREEN (file)	Returns a matrix representing the RGB green component of the BMP, GIF, JPG, or TGA color image in file.
READ_HLS (file)	Returns a packed matrix of hue, lightness, and saturation components based on the Ostwald color model for the BMP, GIF, JPG, or TGA color image in file. The returned matrix contains the H, L, and S matrices packed side by side.
READ_HLS_HUE (file)	Returns a matrix representing the HLS hues based on the Ostwald color model for the BMP, GIF, JPG, or TGA color image in file.
READ_HLS_LIGHT (file)	Returns a matrix representing the HLS lightness components based on the Ostwald color model for the BMP, GIF, JPG, or TGA color image in file.
READ_HLS_SAT (file)	Returns a matrix representing the HLS saturation components based on the Ostwald color model for the BMP, GIF, JPG, or TGA color image in file.
READ_HSV (file)	Returns a packed matrix of hue, saturation, and value components based on Smith's HSV color model for the BMP, GIF, JPG, or TGA color image in file. The returned matrix contains the H, S, and V matrices packed side by side.
READ_HSV_HUE (file)	Returns a matrix representing the HSV hues based on Smith's HSV color model for the BMP, GIF, JPG, or TGA color image in file.
READ_HSV_SAT (file)	Returns a matrix representing the HSV saturation components based on Smith's HSV color model for the BMP, GIF, JPG, or TGA color image in file.
READ_HSV_VALUE (file)	Returns a matrix representing the HSV value components based on Smith's HSV color model for the BMP, GIF, JPG, or TGA color image in file.

READ_IMAGE (file)	Returns a matrix containing a grayscale representation of the BMP, GIF, JPG, or TGA image in file.
READ_RED (file)	Returns a matrix representing the RGB red component of the BMP, GIF, JPG, or TGA image in file.
READBIN (file, type, [[endian], [cols], [skip], [maxrows]])	Returns a matrix from a single-format binary data file of specified type.
READBMP (file)	Returns an array of integers between 0 (black) and 255 (white) representing the grayscale BMP image in file.
READFILE (file, type, [[colwidths], [rows], [cols], [emptyfill]])	Returns a matrix from the contents of a file of specified type (delimited, fixed-width, or Excel).
READPRN (file)	Returns a matrix formed from a structured data file on your file system.
READRGB (file)	Returns a packed matrix of red, green, and blue components for the BMP color image in file. The returned matrix contains the R, G, and B matrices packed side by side.
READWAV (file)	Creates a matrix containing signal amplitudes in file. Each column represents a separate channel of data. Each row corresponds to a moment in time.
regress (vx, vy, n)	Returns a vector of coefficients for the multivariate nth degree least-squares polynomial fit of the data in vx and vy. This vector becomes the first argument of the interp function.
relax (A, B, C, D, E, F, U, rjac)	Returns a square matrix of solution values for Poisson's equation for source function F. Matrices A, B, C, D, and E specify coefficients for linearly approximating the Laplacian operator. Matrix U gives boundary values along the edges. If these are zero, use multigrid.
reverse (A)	Reverses the order of elements in a vector, or of rows in a matrix A.
rexp (m, r)	Returns a vector of m random numbers having the exponential distribution with rate r.
rF (m, d1, d2)	Returns a vector of m random numbers having the F distribution with degrees of freedom d1 and d2.
rgamma (m, s)	Returns a vector of m random numbers having the gamma distribution with shape parameter s.
rgeom (m, q)	Returns a vector of m random numbers having the geometric distribution with probability of success q.
rhypergeom (m, a, b, n)	Returns a vector of m random numbers having the hypergeometric distribution.
Rkadapt (y, x1, x2, npoints, D)	Returns a matrix of solution values for the differential equation specified by the derivatives in D, and having initial conditions y on the interval [x1,x2], using an adaptive step Runge-Kutta method. Parameter npoints controls the number of rows in the matrix output.
rkfixed (y, x1, x2, npoints, D)	Returns a matrix of solution values for the differential equation specified by the derivatives in D and having initial conditions y on the interval [x1,x2] using a fixed step Runge-Kutta method. Parameter npoints controls the number of rows in the matrix output.
rlnorm (m, mu, sigma)	Returns a vector of m random numbers having the lognormal distribution with logmean mu and logdeviation sigma.

rlogis (m, l, s)	Returns a vector of m random numbers having the logistic distribution with location l and scale s.
rnbinom (m, n, p)	Returns a vector of m random numbers having the negative binomial distribution with size n and probability of failure p.
rnd (x)	Returns a uniformly distributed random number between 0 and x.
rnorm (m, mu, sigma)	Returns a vector of m random numbers having the normal distribution with mean mu and standard deviation sigma.
root (f(var), var, [a, b])	Returns the value of var to make the function f equal to zero. If a and b are specified, root finds var on this interval. Otherwise, var must be defined with a guess value before root is called.
round (z, n)	Rounds z to n places. If n is omitted, z is rounded to the nearest integer. If n < 0, z is rounded to the left of the decimal point.
Round (z, y)	Rounds z to the closest multiple of y, typically used for correct unit scaling.
rows (A)	Returns the number of rows in A.
rpois (m, l)	Returns a vector of m random numbers having the Poisson distribution with l > 0.
rref (A)	Returns a matrix representing the row-reduced echelon form of A.
rsort (A, n)	Returns an array formed by rearranging columns of A until row n is in ascending order.
rt (m, d)	Returns a vector of m random numbers having Student's t distribution with d degrees of freedom.
runif (m, a, b)	Returns a vector of m random numbers having the uniform distribution on interval [a,b].
rweibull (m, s)	Returns a vector of m random numbers having the Weibull distribution with shape parameter s.
SaveColormap (file, M)	Creates a colormap named file containing the values in the matrix M. Returns the number of rows written to the file.
sbval (v, x1, x2, D, load, score)	Returns a set of initial conditions for the boundary value problem specified by the derivatives in D and guess values in v on the interval [x1,x2]. Parameter load contains both known initial conditions and guess values from v, and score measures solution discrepancy at x2.
search (S1, SubS, m)	Returns the starting position of the substring SubS in S1 beginning from position m.
sec (z)	Returns the secant of z.
sech (z)	Returns the hyperbolic secant of z.
Seed (x)	Resets the random number seed to x and returns the previous value.
Shi (x)	Returns the value of the hyperbolic sine integral function.
Si (x)	Returns the value of the sine integral function.
sign (x)	Returns 0 if x = 0, 1 if x > 0, and -1 otherwise. For complex values, use csgn.
signum (z, [x])	Returns x if z = 0 and z/|z| otherwise.
sin (z)	Returns the sine of z.
sinc (z)	Returns the value of sin(z)/z, with correct behavior in the limit as z approaches 0.

sinfit (vx, vy, vg)	Returns a vector containing the coefficients for a sine curve of the form a*sin(x +b) + c that best approximates the data in vectors vx and vy. Vector vg contains guess values for the three coefficients.
sinh (z)	Returns the hyperbolic sine of z.
SIUnitsOf (x)	Returns the units of x. If x has no units, returns 1.
skew (A, B, C, ...)	Returns the skewness of the elements of A, B, C, ...
slope (vx, vy)	Returns the slope of line that best fits data in vx and vy.
sort (v)	Returns a vector with the values from v sorted in ascending order.
sph2xyz (r, theta, phi)	Converts the spherical coordinates of a point in 3D space to rectangular coordinates.
stack (A, B, C, ...)	Returns an array formed by placing A, B, C, ... top to bottom. A, B, C, ... are arrays having the same number of columns, or they are scalars and column vectors.
statespace (init, t1, t2, npoints, A, [B], [u])	Returns the solution to a system of linear, first-order ordinary differential equations.
stderr (vx, vy)	Returns the standard error associated with a linear regression for the points described by the vectors vx and vy. Measures the spread of data points about the regression line.
stdev (A, B, C, ...)	Returns the population standard deviation of the elements of A, B, C, ...
Stdev (A, B, C, ...)	Returns the sample standard deviation of the elements of A, B, C ...
Stiffb (y, x1, x2, npoints, D, AJ)	Returns a matrix of solution values for the stiff differential equation specified by the derivatives in D, the augmented Jacobian function AJ, and initial conditions y on the interval [x1,x2] using a Bulirsch-Stoer method. Parameter npoints controls the number of rows in the matrix output.
Stiffr (y, x1, x2, npoints, D, AJ)	Returns a matrix of solution values for the stiff differential equation specified by the derivatives in D, the augmented Jacobian function AJ, and initial conditions y on the interval [x1,x2] using a Rosenbrock method. Parameter npoints controls the number of rows in the matrix output.
str2num (S)	Returns a constant formed by converting string S into a number.
str2vec (S)	Returns a vector of ASCII codes corresponding to the characters in S.
strlen (S)	Returns the number of characters in string S.
submatrix (A, ir, jr, ic, jc)	Returns the submatrix of array A consisting of elements in rows ir through jr and columns ic through jc of A.
substr (S, m, n)	Returns a substring of S beginning at character m and having maximum length n.
supsmooth (vx, vy)	Returns a vector created by the piecewise use of a symmetric nearest neighbor linear least-squares fitting on each element in vy, in which the number of nearest neighbors is adaptively chosen.
svd2 (A)	Returns a vector of 3 nested arrays. The first array contains the vector of singular values. The following two arrays are the matrices U and V.

tan (z)	Returns the tangent of z.
tanh (z)	Returns the hyperbolic tangent of z.
Tcheb (n, x)	Returns the value of the Chebyshev polynomial of degree n, of the first kind, at x.
time (z)	Returns the current system time. The value z is an arbitrary Mathcad expression with no impact on the return.
tr (M)	Returns the trace of square matrix M: sum of diagonal elements.
trace (S, x, y, z, ...)	Returns a string containing the value of the arguments x, y, z, ... with print order and surrounding text specified by S. Prints values in the Trace Window when debug mode is on. S is optional if only one value is printed.
trunc (z)	Returns the integer part of z by removing the fractional part.
Trunc (z, y)	Returns the value of trunc(z/y)*y, typically used for correct unit scaling.
Ucheb (n, x)	Returns the value of the Chebyshev polynomial of degree n, of the second kind, at x.
until (icond, x)	Returns x until icond is negative.
var (A, B, C, ...)	Returns the population variance of the elements of A, B, C, ...
Var (A, B, C, ...)	Returns the sample variance of the elements of A, B, C, ...
vec2str (v)	Returns a string formed by converting the ASCII codes in v to characters.
vlookup (z, A, c)	Looks in the first column of a matrix, A, for a given value, z, and returns the value(s) in the same row(s) in the column specified, c. When multiple values are returned, they appear in a vector.
wave (v)	Returns the one-dimensional discrete wavelet transform of the data in v using the Daubechies 4-coefficient wavelet filter.
WRITE_HLS (file)	Writes a packed matrix consisting of the hue, lightness, and saturation components of an image to a 16 million color Windows BMP file on your file system.
WRITE_HSV (file)	Writes a packed matrix consisting of the hue, saturation, and value components of an image to a 16 million color Windows BMP file on your file system.
WRITEBIN (file, type, endian)	Writes an array of scalars to the binary data file named file.
WRITEBMP (file)	Writes an array into a grayscale BMP file on your file system.
WRITEPRN (file)	Writes an array into a file on your file system.
WRITERGB (file)	Writes a packed matrix consisting of the red, green, and blue components image to a 16 million color Windows BMP file on your file system.
WRITEWAV (file, s, b)	Writes a WAV signal file out of a matrix.
xy2pol (x, y)	Converts the rectangular coordinates of a point in 2D space to polar coordinates.
xyz2cyl (x, y, z)	Converts the rectangular coordinates of a point in 3D space to cylindrical coordinates.
xyz2sph (x, y, z)	Converts the rectangular coordinates of a point in 3D space to spherical coordinates.
Y0 (z)	Returns the zeroth order Bessel function of the second kind.

Y0.sc (z)	Returns the zeroth order Bessel function of the second kind, scaled by the factor exp(-|Im(z)|).
Y1 (z)	Returns the first order Bessel function of the second kind.
Y1.sc (z)	Returns the first order Bessel function of the second kind, scaled by the factor exp(-|Im(z)|).
Yn (m, z)	Returns the mth order Bessel function of the second kind.
Yn.sc (m, z)	Returns the mth order Bessel function of the second kind, scaled by the factor exp(-|Im(z)|).
ys (m, z)	Returns the value of the spherical Bessel function of the second kind, of order m.
Zeta (s)	Returns the value of the Riemann Zeta function. Can only be evaluated symbolically.

Mathcad Data Analysis Extension Pack Functions

Function Name (Parameters)	Function Definition
Bicubic2D (vx, vy, Z, p, q)	Interpolates between 2-D values in Z, with locations in vx and vy, at intermediate point (p, q).
Binterp (x, b)	Interpolates results b from Spline2 (b) at point x, along with the first, second, and third derivatives.
confidence (vx, vy, F, b, conf)	Returns the confidence limits on the parameters b of a fitting function F(x,b) fit to the data vx and vy.
contingtbl (M)	Returns chi-squared, degrees of freedom, probability that chi-squared or larger would occur if variables had no association, Cramer's V, and the contingency coefficient, C for a contingency table M.
DWS (b)	Returns the Durbin-Watson statistic for the result vector, b, of the Spline2 function.
filterNaN (v)	Removes the rows of the data set, v, that have NaNs.
Ftest (v1, v2)	Tests the hypothesis that v1 and v2 are drawn from distributions having the same variance. Returns the F statistic, and the probability that a value this large or larger would occur when the distributions have the same variance.
Grubbs (v, a)	Returns indices of suspected outliers, and their Grubbs test statistics for a confidence level a.
GrubbsClassic (v, a)	Returns index of the data point most likely to be an outlier, and its Grubbs test statistic for a confidence level a.
Hlookup (z, A, r,"modifier")	Looks in the first row of A for values matched by z according to the boolean modifier. Returns the value(s) in the matched column(s) in row r.
kendltau (v1, v2)	Returns Kendall's tau, number of standard deviations from 0, and the probability that a value this large or larger would occur if the samples were uncorrelated.
kendltau2 (M)	Returns Kendall's tau, number of standard deviations from 0, and the probability that a value this large or larger would occur if contingency table M were uncorrelated.
LeastSquaresFit (vx, vy, F, guess, conf, [Stdy], [LBUB],[Acc])	Returns parameters and their confidence limits for the nonlinear fitting function F for the data vx and vy, for a confidence level conf, with optional standard deviations Stdy and optional lower and upper bounds on acceptable parameter values to accuracy Acc.

loadings (nipals)	Returns the loadings of the principal components (eigenvalues) from multivariate data returned by the nipals function.
localmax (data, [w])	Returns the local maxima in data by nearest neighbor comparison, with an optional window width w of comparison points.
localmin (data, [w])	Returns the local minima in data by nearest neighbor comparison, with an optional window width w of comparison points.
Lookup (z, A, B, modifier)	Looks in the matrix A for values matched by z according to the boolean modifier. Returns the value(s) in the same position(s) in matrix B.
markNaN (data, vindex)	Changes each element in data specified by vindex to contain a NaN.
Match (z, A, "modifier")	Returns the indices of entries in A which match z according to the boolean modifier.
matchNaN (data)	Returns the index or pair of indexes of the NaN entries in data.
Nipals (Data, numPC, maxiter, ["scale"])	Returns numPC principal components (eigenvalues), loadings, scores, and accumulated variance explained by each PC from multivariate data using a maximum of maxiter iterations. TOL specifies the termination accuracy used for eigenvalue generation. Data may be optionally scaled to the standard deviation.
Nipals2 (nipals, numPC)	Returns numPC additional principal components (eigenvalues), loadings, scores, and accumulated variance explained by each PC given the results calculated by Nipals.
order (v)	Returns the index in which the entries of v occur if sorted, based on the current value of ORIGIN.
PCAeigenvals (nipals)	Returns the principal components (eigenvalues) from multivariate data returned by the nipals function.
PCAvariance (nipals)	Returns the accumulated percentage of variance explained by the calculated principal components (eigenvalues) returned by the nipals function.
percentile (v, p)	Returns the number of values in v below p percent of the total number of points.
polycoeff (vx, vy)	Returns the coefficients of the interpolating polynomial function.
polyint (vx, vy, x)	Returns interpolated value at x using a polynomial function, and the expected error.
polyiter (vx, vy, x, N, e)	Returns interpolated value at x using a polynomial function with maximum order N and maximum error, e. Also returns the calculated error, and whether the function converged.
qqplot (v1, [v2 or "distrib"])	Returns points on a probability plot. If only v1 is specified, quantiles for v1 and the normal distribution are returned. If v2 is specifed, quantiles for v1 and v2 are returned. If 'weibull' is specified, returns natural log quantiles for v1 and the weibull quantiles.
Rank (v)	Returns the averaged position at which each value in v appears in a sorted list of the data.

rationalfit (vx, vy, conf, [m, n], [resid], [Stdy], [LBUB])	Returns parameters and their confidence limits for a rational polynomial fit of order m and n on the top and bottom, or an allowable residual chi-squared, if the function should determine the optimal order. Confidence level conf is achieved, with optional standard deviations Stdy and optional lower and upper bounds on acceptable parameter values.
rationalfitnp (vx, vy, conf, [m, n], [resid], [Stdy], [LBUB])	Returns parameters and their confidence limits for a rational polynomial fit of order m and n on the top and bottom, or an allowable residual chi-squared, if the function should determine the optimal order. Confidence level conf is achieved, with optional standard deviations Stdy and optional lower and upper bounds on acceptable parameter values.
rationalint (vx, vy, x)	Returns interpolated value at x using rational functions, and the expected error.
Scale (M, min, max)	Scales all values in M between min and max.
scores (nipals)	Returns the scores of the principal components (eigenvalues) from multivariate data returned by the nipals function.
Spear (v1, v2)	Returns Spearman's rank correlation coefficient, and associated statistics.
Spline2 (vx, vy, n, [vw], [u], [level])	Returns the optimal set of order-n B-spline knots to interpolate on data vx and vy, with optional weights vw, optional desired knots u, and an optional reject level. Output is used with Binterp.
Thiele (vx, coeff, x)	Returns the interpolated y value for the real scalar x, using the data points in vx and the coefficients returned by Thielecoeff.
Thielecoeff (vx,vy)	Returns the continued fraction coefficients of the vectors vx and vy.
ThreeSigma (v)	Returns indices of points in v whose mean divided by standard deviation is greater than three (outlier test), and the value of this quantity for each point.
trim (vdata, vindex)	Trims out the entries (rows) specified by vindex.
vhlookup (z1, z2, A)	Looks in the first column and row of A for values matched by z1 and z2, respectively. Returns the value(s) in the intersection of matched rows and columns.
VHlookup (z1, z2, A, "modifier")	Looks in the first column and row of A for values matched by z1 and z2 according to the boolean modifier. Returns the value(s) in the intersection of matched rows and columns.
Vlookup (z, A, c, "modifier")	Looks in the first column of a matrix, A, for values matched by z according to the boolean modifier. Returns the value(s) in the matched row(s) in column c. When multiple values are returned, they appear in a vector.
VSmooth (v, w)	Repeatedly median smoothes v until no additional change has occurred for each window width in w.

Mathcad Image Processing Extension Pack Functions

Function Name (Parameters)	Function Definition
addnoise (M, p, n)	Returns matrix M with added noise, where the noise has probabilty p/2 to add n to a pixel, and p/2 to subtract n.

and (M, N)	Returns the boolean AND of two image matrices M and N, which must be the same size.
augment3 (X, Y, Z)	Returns a matrix formed by putting matrices (or vectors) X, Y, and Z side by side. They must all have the same number of rows.
binarize (M, thresh)	Returns a binarized version of matrix M with pixels above threshold thresh set to 1 and below to 0.
binarize_auto (M)	Returns a binarized version of matrix M, choosing the threshold automatically.
binarize2 (M, lowThresh, highThresh, inValue, outValue)	Returns a binarized version of matrix M with pixels between lowThresh and highThresh set to inValue, and pixels outside to outValue.
blend (M, N)	Returns a blend of same-size matrices M and N (pixelwise sum - [product/255]).
canny (M, sigma, low, high)	Returns a binary edge image resulting from Canny edge detection on matrix M, using standard deviation sigma and hysteresis thresholds low and high.
center (M)	Returns fourier transform image matrix M tansformed so that DC is in the center.
centsmooth (M)	Returns matrix M smoothed with a 3x3 center weighted kernel.
clip (M, Min, Max)	Returns matrix M with elements clipped to lie between Min and Max.
close (M, Melem, b)	Performs binary closing on matrix M at threshold b with structuring element Melem.
cnvxhull (M, fg)	Returns a matrix containing the convex hull of pixels of value fg in matrix M.
colgrad (M)	Returns the column gradient (difference by columns) of matrix M.
compgrad (M)	Performs edge detection by comparing the gradients of the 8 neighbors on matrix M.
concomp (M, con, fg)	Performs connected component labeling of the pixels with grayscale value fg in matrix M, considering 4-connected neighbors if con is 4 or 8-connected if con is 8.
convol2d (M, K)	Returns the convolution of matrix M with kernel K.
convolve3 (M, K)	Returns the quick convolution of matrix M with 3x3 kernel K.
convolve5 (M, K)	Returns the quick convolution of matrix M with 5x5 kernel K.
dct2d (M)	Returns the 2D discrete cosine transform (type II) of matrix M.
diacrisp (M)	Returns matrix M crisped with a 3x3 diagonally weighted kernel.
difedge (M)	Performs edge-detection by differential convolution on matrix M.
dilate (M, Melem, r_origin, c_origin, b)	Performs binary dilation on matrix M at threshold b using structuring element Melem with origin at row r_origin and column c_origin.
dilate4 (M, b)	Performs dilation on matrix M at threshold b using 4 neighbors.
dilate8 (M, b)	Performs dilation on matrix M at threshold b using 8 neighbors.

distform (M, fg)	Returns the distance transform of image M using foreground gray value fg.
equalize (M)	Returns matrix M with grayscale adjusted to form a linear cumulative histogram.
erode (M, Melem, r_origin, c_origin, b)	Performs binary erosion on matrix M at threshold b using structuring element Melem with origin at row r_origin and column c_origin.
erode4 (M, b)	Performs erosion on matrix M at threshold b using 4 neighbors.
erode8 (M, b)	Performs erosion on matrix M at threshold b using 8 neighbors.
extract (M, n)	Returns the nth (1, 2, or 3) color component of packed 3-color matrix M.
freichen (M)	Performs edge detection by Frei-Chen convolution on matrix M.
funcdeconv (M, f, e)	Deconvolution of matrix M with frequency domain function f and error e.
funconv (M, f)	Convolution of matrix M with frequency domain function f.
funmap (M, f)	Returns matrix M with function f applied to each element.
gaussconv (M, s)	Convolution of matrix M with frequency domain gaussian of half-width s.
gaussdeconv (M, s, e)	Deconvolution of matrix M with frequency domain gaussian of half-width s with error e.
getnoise (M)	Returns the difference between matrix M and median filtered M.
gray_close (M, Melem)	Performs grayscale closing on matrix M with structuring element Melem.
gray_dilate (M, Melem, r_origin, c_origin)	Performs grayscale dilation on matrix M using structuring element Melem with origin at row r_origin and column c_origin.
gray_erode (M, Melem, r_origin, c_origin)	Performs grayscale erosion on matrix M using structuring element Melem with origin at row r_origin and column c_origin.
gray_open (M, Melem)	Performs grayscale opening on matrix M with structuring element Melem.
gray_to_rgb (gray, colormap)	Returns grayscale matrix gray converted to color using color palette matrix colormap.
hist2d (M, N, n)	Returns a two-dimensional histogram with n bins on equal-sized matrices M and N.
hls_to_rgb (HLS)	Returns array HLS in HLS color representation converted to RGB color representation.
horzflip (M)	Returns the matrix M flipped horizontally.
hsv_to_rgb (HSV)	Returns array HSV in HSV color representation converted to RGB color representation.
idct2d (M)	Returns the inverse 2D discrete cosine transform (type II) of matrix M.
imhist (M, n)	Returns an n-bin histogram of M for values between 0 and 255 (ignores values outside that range).
imhist2 (M, n)	Returns an n-bin histogram of M over its range of values.
immse (M, Q)	Returns the mean-squared-error (MSE) between image matrices M and Q.

imquant (M, n)	Returns a quantized version of matrix M containing only n equally-spaced grayscale levels between 0 and 255.
imquant2 (M, v)	Returns a quantized version of matrix M containing only the grayscale levels in vector v.
imsnr (M, Q)	Returns the signal-to-noise ratio (SNR) between image matrices M and Q.
invert (M)	Returns the matrix M with each element set to 255 - element.
invert2 (M)	Returns the matrix M with each element set to max(M) - element + min(M).
iwave2d (M, n)	The n-level inverse wavelet transform of M.
kirsch (M)	Performs edge detection by kirsch convolution and comparison on matrix M.
laplace24 (M)	Returns the convolution of matrix M with a 5x5 Laplacian kernel. The kernel's center is 24.
laplace4 (M)	Returns the convolution of matrix M with a 3x3 Laplacian kernel. The kernel's center is 4.
laplace8 (M)	Returns the convolution of matrix M with a 3x3 Laplacian kernel. The kernel's center is 8.
levelmap (M, vec)	Returns matrix with values in vec assigned by matching vec's indices to elements in matrix M. vec must be such that the elements of M are between 0 and length(vec) - 1.
mask (M, N)	Returns matrix M masked by same-size matrix N (i.e. with zeros where N is zero).
matconv (M, N)	Convolution of matrix M with frequency domain mask N.
matdeconv (M, N, e)	Deconvolution of matrix M with frequency domain mask N and error e.
medfilt (M)	Returns median filtered M.
moment_invariant (M)	Returns a vector containing the seven typical moment invariants of M.
open (M, Melem, b)	Performs binary opening on matrix M at threshold b using structuring element Melem.
or (M, N)	Returns boolean OR of two image matrices M and N, which must be the same size.
orthocrisp (M)	Returns matrix M crisped with a 3x3 orthogonally weighted kernel.
orthocrisp5 (M)	Returns matrix M crisped with a 5x5 orthogonally weighted kernel.
orthosmooth (M)	Returns matrix M smoothed with a 3x3 orthogonally weighted kernel.
orthosmooth5 (M)	Returns matrix M smoothed with a 5x5 orthogonally weighted kernel.
prewitt (M)	Performs edge detection by Prewitt convolution on matrix M.
putregion (M, N, row, col)	Returns the matrix N inserted into M at row row and column col.
quantfilt (M, elem, quantile)	Performs quantile filtering on M using neighborhood matrix elem and quantile probability quantile.

READRAW (filename, rows, cols, bits, endian, skip)	Returns the contents of a raw binary image file as a matrix. The binary file is interpreted to contain a rows x cols matrix of bits (8 or 16) bits per pixel packed integers, in "Little" or "Big" endian format, and skip bytes are skipped for header at the beginning of the file.
reg_grow (M, x_gridsize, y_gridsize, num_regions)	Performs the piecewise-constant energy-based region growing segmentation of M into num_regions regions, using initial grid spaced by x_gridsize along x and y_gridsize along y.
relerror (M, Q)	Returns the relative error between matrices M and Q.
replace (M, N, n)	Returns packed image matrix M with the nth (1, 2, or 3) color component replaced by matrix N, which must have the same number of rows as M and 1/3 as many columns.
rgb_to_gray (RGB)	Returns RGB color array RGB converted to grayscale.
rgb_to_hls (RGB)	Returns array RGB in RGB color representation converted to HLS color representation.
rgb_to_hsv (M)	Returns array RGB in RGB color representation converted to HSV color representation.
rgb_to_ycbcr (RGB)	Returns array RGB in RGB color representation converted to YCbCr color representation.
rgb_to_yiq (RGB)	Returns array RGB in RGB color representation converted to YIQ color representation.
roberts (M)	Performs edge detection by Roberts convolution on matrix M.
robinson3 (M)	Performs edge detection by 3x3 Robinson convolution and comparison on matrix M.
robinson5 (M)	Performs edge detection by 5x5 Robinson convolution and comparison on matrix M.
rotate (M, angle)	Returns the matrix M rotated angle degrees counterclockwise.
rotate180 (M)	Returns the matrix M rotated 180 degrees counterclockwise.
rotate270 (M)	Returns the matrix M rotated 270 degrees counterclockwise.
rotate90 (M)	Returns the matrix M rotated 90 degrees counterclockwise.
rowgrad (M)	Returns the row gradient (difference by rows) of matrix M.
scale (M, Min, Max)	Returns matrix M with elements scaled between Min and Max.
shape_features (M)	Returns a matrix of moments and shape features for each distinct pixel value in labeled image M.
skeleton (B)	Returns binary matrix B eroded to its innermost level.
skeleton2 (M, b)	Returns the skeleton of matrix M binarized with threshold b.
sobel (M)	Performs edge detection by Sobel convolution on matrix M.
subcolor (M, ir, jr, ic, jc)	Returns the submatrix from row ir to jr, column ic to jc, of packed color matrix M.
thin (M, b)	Returns the thinned version of matrix M binarized with threshold b.
threshold (M, thresh)	Returns the matrix M with every element below thresh set to thresh. If thresh is negative, every element above -thresh is set to -thresh.
translate (M, rows, cols, pad)	Returns matrix M translated by rows rows and cols colums, padding unfilled matrix elements with pad.

unicrisp (M)	Returns matrix M crisped with a 3x3 uniformly weighted kernel.
unismooth (M)	Returns matrix M smoothed with a 3x3 uniformly weighted kernel.
unismooth5 (M)	Returns matrix M smoothed with a 5x5 uniformly weighted kernel.
vertflip (M)	Returns the matrix M flipped vertically.
warp (M, T)	Performs bilinear warping on matrix M, using tie-points stored in matrix T.
wave2d (M, n)	The n-level wavelet tranform of M.
wavescale (M, n, Min, Max)	The n-level wavelet transform of M, scaled between Min and Max.
wiener2d (M, win_h, win_w)	Perform 2D adaptive Wiener filtering on M using a local window win_w pixels wide by win_h pixels high.
WRITERAW (filename, bits, endian)	Writes a matrix M to raw binary integer image file filename, using either 8 or 16 bits per pixel, in "Little" or "Big" endian storage order. Set this function equal to the matrix M.
ycbcr_to_rgb (YCbCr)	Returns array YCbCr in YCbCr color representation converted to RGB color representation.
yiq_to_rgb (YIQ)	Returns array YIQ in YIQ color representation converted to RGB color representation.
zoom (M, hscale, vscale)	Return image matrix M resized by factor hscale horizontally and vscale vertically.

Mathcad Signal Processing Extension Pack Functions

Function Name (Parameters)	Function Definition
bandpass (f1, f2, n, [w])	Returns coefficients for a bandpass FIR filter with n coefficients and cutoff frequencies f1 and f2, windowed with a taper w.
bandstop (f1, f2, n, [w])	Returns coefficients for a bandstop FIR filter with n coefficients and cutoff frequencies f1 and f2, windowed with taper w.
bessel (n, scale)	Returns coefficients for an analog Bessel filter of order n; scale controls the gain at cutoff. The output is an argument to the functions iirlow, iirhigh, iirpass or iirstop.
blackman (n)	Returns a Blackman window of width n.
burg (v, n)	Returns coefficients for nth order linear prediction generated from the vector v using Burg's method.
butter (n)	Returns coefficients for an analog Butterworth filter of order n. The output is used as an argument to one of the functions iirlow, iirhigh, iirpass or iirstop.
ccepstrum (A)	Returns a matrix containing the complex cepstrum of a multichannel signal A.
cepstrum (v)	Returns the cepstrum of the vector v.
cheby (n, b)	Returns a Chebyshev window of width n and parameter b.
cheby1 (n, e)	Returns coefficients for a Type I Chebyshev analog filter of order n; ripple is controlled by the parameter e. The output is used as an argument to one of the functions iirlow, iirhigh, iirpass or iirstop.

cheby2 (n, scale, atten)	Returns coefficients for a Type II Chebyshev analog filter of ord. n with the lower edge of the stopband at scale and stopband attenuation atten. The output is an argument to the functions iirlow, iirhigh, iirpass or iirstop.
chirpz (v, lo, hi, d)	Returns the frequency spectrum of the signal v between lo and hi at frequency intervals of d.
coherence (vx, vy, n, r, [w])	Returns the coherence of vectors vx and vy. The signal vectors are divided into n overlapping intervals with fraction of overlap r. Each data segment is windowed with taper w.
convol (vx, vy)	Returns the convolution of the arrays vx and vy.
costaper (n, a)	Returns a cosine taper window of length n, with the percentage of raised cosine given by a.
costr (v)	Returns the cosine transform of the array v.
covar (vx, vy)	Returns the covariance of the arrays vx and vy.
cspectrum (vx, vy, n, r, [w])	Returns the cross spectrum of the vectors vx and vy. The signal vectors are divided into n overlapping intervals with fraction of overlap r. Each data segment is windowed with taper w.
deconvol (vz, vx)	Returns the deconvolution of the array vz by the array vx.
detrend (v)	Returns the vector or matrix v with any linear trend removed.
dht (v)	Returns the Hartley transform of the array v.
dwavelet (v)	Returns a discrete wavelet transform of the array v.
expsmooth (v, a)	Returns a smoothed version of data in array v generated by exponential smoothing with weight a.
fftfilt (v, C, [nfft])	Returns a matrix with the result of the convolution of a multichannel signal v, with an array C, calculated by computing the FFT of length nfft.
gain (C, f)	Returns the gain at frequency f of a filter with coefficients C.
gaussian (n, a)	Returns a Gaussian window of width n; the parameter a controls the peak width.
gaussn (n)	Returns an n-element vector of random numbers following a Gaussian probability distribution of mean 0 and standard deviation 1.
hamming (n)	Returns a Hamming window of width n.
hanning (n)	Returns a Hanning window of width n.
highpass (f, n, [w])	Returns coefficients for a highpass FIR filter with n coefficients and cutoff frequency f, windowed with taper w.
hilbert (v)	Returns the Hilbert transform of the array v.
icostr (v)	Returns the inverse cosine transform of the array v.
idht (v)	Returns the inverse Hartley transform of the array v.
idwavelet (v)	Returns the inverse of the discrete wavelet transform carried out by the dwavelet function.
iirhigh (C, f)	Returns coefficients for a highpass IIR filter with cutoff frequency f. The input array C is generated by one of the functions cheby1, cheby2, bessel, or butter.
iirlow (C, f)	Returns coefficients for a lowpass IIR filter with cutoff frequency f. The input array C is generated by one of the functions cheby1, cheby2, bessel, or butter.
iirpass (C, lowf, highf)	Returns coefficients for a bandpass IIR filter with cutoff frequencies lowf and highf. The input array C is generated by one of the functions cheby1, cheby2, bessel, or butter.

iirstop (C, lowf, highf)	Returns coefficients for a bandstop IIR filter with cutoff frequencies lowf and highf. The input array C is generated by one of the functions cheby1, cheby2, bessel, or butter.
interpolate (v, n)	Returns an interpolated version of the vector v with n times as many points as v.
isintr (v)	Returns the inverse sine transform of the array v.
kaiser (n, b)	Returns a Kaiser window of width n and parameter b.
lcorr (vx, vy)	Returns a vector giving the correlation of the vectors vx and vy at each possible lag.
lowpass (f, n, [w])	Returns coefficients for a lowpass FIR filter with n coefficients and cutoff frequency f, windowed with taper w.
mag (v)	Returns a vector containing the magnitudes of the elements in the complex array v.
makeri (magvec, phasevec)	Returns a vector of complex numbers whose magnitudes and phases are contained in magvec and phasevec.
medfilt1d (M, n)	Returns the multichannel signal M, filtered with a median filter of length n.
movavg (v, n)	Returns a smoothed version of data in array v created by taking a moving average with a window of width n.
multirate (v, n, m, [f])	Returns a version of multichannel signal v resampled by a factor of n/m, using the optional argument f as the lowpass interpolating filter.
nuttall (n)	Returns a Nuttall window of width n.
onefn (n)	Returns an n-element vector of 1/f noise.
phase (v)	Returns a vector containing the phases of the elements in the complex array v.
phasecor (phasevec)	Returns a vector generated from the array of phases phasevec by removing jump discontinuities.
plcorr (v)	Returns a vector giving the partial autocorrelation of the vector v at each lag.
pspectrum (v, n, r, [w])	Returns the power spectrum of v, computed by dividing v into n overlapping segments with overlap fraction r. Each data segment is windowed with taper w.
quantize (v, n)	Returns a vector that assigns the elements of v to n equally spaced levels.
recenter (A)	Returns a recentered version of array A, cyclically permuted to bring the first element of A to the middle.
remez (vg, vr, vw, n)	Returns coefficients for an FIR filter of length n generated by the Remez exchange algorithm.
resample (v, m, n)	Returns a vector obtained from v by n-fold interpolation and then sampling every mth point.
response (v, C, n)	Returns a vector giving n terms of the response of an input vector v to an FIR filter with coefficient array C.
sintr (v)	Returns the sine transform of the array v.
snr (vx, vy, n, r, [w])	Returns the signal-to-noise ratio for vx and vy. The signal vectors are divided into n overlapping segments with fraction of overlap r. Each data segment is windowed with taper w.
stft (v, [n], [s], [w])	Returns a matrix containing the short time Fourier transform of a signal v; computes n frequencies, every s samples, and tapers each block with window (or window type) w.
taprect (n)	Returns a tapered rectangular window of width n.

timecorr (v, t, l, s, [d])	Returns a matrix containing the samples of a time-depemdent autocorrelation function for a signal v, type t, l number of lags, step size s, and an optional scalar or matrix argument d.
timefreq (v, t, f, s, [d])	Returns a matrix containing the samples of a time-depemdent autocorrelation function for a signal v, type t, f number of frequencies, step size s, and an optional scalar or matrix argument d.
triangular (n)	Returns a triangular window of width n.
twodconvol (A, B)	Returns the two-dimensional convolution of the arrays A and B.
walsh (n, k)	Returns the kth Walsh function of order n (n between 0 and 12).
whiten (n)	Returns an n-element vector of uniformly distributed white noise.
yulew (v, n)	Returns coefficients for nth order linear prediction generated from the vector v using the Yule-Walker algorithm.

Mathcad Wavelets Extension Pack Functions

Function Name (Parameters)	Function Definition
applybs (w, b, f)	Returns the result of applying the function f to each subband of the wavelet packet basis b comprising the coefficients in w. b is a 2- or 3-column matrix representing a packet basis.
applytbl (T, f)	Returns a new wavelet packet table by applying the function f to each subband of the wavelet packet table T.
basis_display2d (b)	Creates a matrix for graphically displaying 2-dimensional basis b.
best_basis (T, f)	Returns the best packet basis from a packet table T, based on the additive cost function f.
bl (n)	Returns the Battle-Lemarie wavelet based upon the degree n cardinal B-spline. n is an integer bewtween 0 and 6.
Bspline (m, n)	Returns the analysis and synthesis low-pass filters for a spline-based biorthogonal wavelet. m is 1, 2, or 3. When m=1, n is an odd number between 1 and 5. When m=2, n is an even number between 2 and 8. When m=3, n is an odd number between 1 and 9.
coiflet (n)	Returns the low-pass filter of the n-coefficient Coiflet wavelet. n is 6, 12, 18, 24, or 30.
cpt (a, n)	Returns the n-level local cosine packet table for the data array a, using the default trig data taper function.
create_level (n, d)	Returns a d-dimensional wavelet packet basis of all wavelet packets at the level n.
daublet (n)	Returns the low-pass filter of the n-coefficient minimum phase Daubechies wavelet. n is an even integer between 2 and 20.
dlfather (x, [j, k], s)	Returns the value at x of the dual father wavelet packet based on the filter represented by string s at scale j and position k. If j and k are omitted, they are assumed to be 0.
dlfather2d (x, y, [j, kx, ky], s)	Returns the value at (x,y) of the 2D dual father wavelet packet based on the filter represented by string s at scale j and position (kx,ky). If j, kx, and ky are omitted, they are assumed to be 0.

dlmother (x, [j, k], s)	Returns the value at x of the dual mother wavelet packet based on the filter represented by string s at scale j and position k. If j and k are omitted, they are assumed to be 0.
dlmother2d (x, y, [j, kx, ky], s, ds)	Returns the value at (x,y) of the 2D dual mother wavelet packet based on the filter represented by string s at scale j and position (kx,ky). String ds is either "H", "V", or "D", and indicates either the horizontal, vertical, or diagonal mother wavelet. If j, kx, and ky are omitted, they are assumed to be 0.
dlwavelet (x, j, k, n, f, [o, [i]])	Returns the value at x of the nth dual wavelet packet based on the filter f at scale j and position k. f is either a vector or a 2-column matrix. f can have an offset o and negation i.
dlwavelet2d (x, y, j, kx, ky, m, n, f, [o, [i]])	Returns the value at (x,y) of the (m,n)th dual wavelet packet based on filter f at scale j and position (kx,ky). f is either a vector or a 2-column matrix. f can have an offset o and negation i.
dwt (a, n, f, [o, [i]])	Returns the n-level wavelet transform of the array a by the wavelet filter f. f is either a vector or a 2-column matrix. f can have an offset o and negation i.
dwti (a, n, f)	Returns the n-level interval wavelet transform of the array a by the wavelet filter f. f is a vector returned by daublet or symmlet.
dwts (a, n, f)	Returns the n-level symmetric wavelet transform of the array a by the wavelet filter f. f is either a vector or a 2-column matrix returned by bl, Bspline, or daublet(2).
evalbs (w, b, f)	Returns the result of applying the function f to each subband of the wavelet packet basis b comprising the coefficients in w. b is a 2- or 3-column matrix representing a packet basis.
evaltbl (T, f)	Returns the result of applying the function f to each subband of the wavelet packet table T.
extract_basis (T, b)	Returns coefficients in the basis specified by b from a 1D or 2D packet table T.
father (x, [j, k], s)	Returns the value at x of the father wavelet packet based on the filter represented by string s at scale j and position k. If j and k are omitted, they are assumed to be 0.
father2d (x, y, [j, kx, ky], s)	Returns the value at (x,y) of the 2D father wavelet packet based on the filter represented by string s at scale j and position (kx,ky). If j, kx, and ky are omitted, they are assumed to be 0.
get_detail (w, l)	Returns the result of extracting the index-1 block of wavelet packet coefficients at level l from the wavelet packet transform w.
get_detail2d (M, l, ds)	Returns the result of extracting the subband indicated by the string ds of wavelet packet coefficients at level l from the 2-dimensional wavelet packet transform M. ds is either "H", "V", or "D", and indicates either the index-(1,0), index-(0,1), or index-(1,1) subband.
get_smooth (w, l)	Returns the result of extracting the 0th block of wavelet packet coefficients at level l from the wavelet packet transform w.

get_smooth2d (M, l)	Returns the result of extracting the (0,0)th subband of wavelet packet coefficients at level l from the 2-dimensional wavelet packet transform M.
get_subband (w, l, n)	Returns the nth block of wavelet packet coefficients at level l extracted from the wavelet packet transform w. w is a vector.
get_subband2d (M, l, m, n)	Returns the result of extracting the (m,n)th subband of wavelet packet coefficients at level l from the 2-dimensional wavelet packet transform M. M is an array.
icpt (w, n, b)	If w is an array of cosine packet coefficients constructed from a cosine packet table with n levels, returns the inverse cosine packet transform with respect to the basis specified by b, using the default trig data taper function. b is a 2- or 3-column matrix representing a cosine packet basis.
idwt (a, n, f, [o, [i]])	Returns the n-level inverse wavelet transform of the array a by the wavelet filter f. f is either a vector or a 2-column matrix. f can have an offset o and negation i.
idwti (a, n, f)	Returns the n-level inverse interval wavelet transform of the array a by the wavelet filter f. f is a vector returned by daublet or symmlet.
idwts (a, n, f)	Returns the n-level inverse symmetric wavelet transform of the array a by the wavelet filter f. f is either a vector or a 2-column matrix returned by bl, Bspline, or daublet(2).
ilct (w, n, b, f)	If w is an array of cosine packet coefficients constructed from a cosine packet table with n levels, returns the inverse cosine packet transform with respect to the basis specified by b, using data taper function f. b is a 2- or 3-column matrix representing a cosine packet basis.
iwpt (w, b, f, [o, [i]])	Returns the inverse wavelet packet transform of the array w with respect to basis b by the wavelet filter f. b is a 2- or-3-column matrix representing a packet basis. f is either a vector or a 2-column matrix. f can have an offset o and negation i.
iwpti (w, b, f)	Returns the inverse interval wavelet packet transform of the array w with respect to basis b by the wavelet filter f. b is a 2- or 3-column matrix representing a packet basis. f is a vector.
iwpts (w, b, f)	Returns the inverse symmetric wavelet packet transform of the array w with respect to basis b by the wavelet filter f. b is a 2- or 3-column matrix representing a packet basis. f is either a vector or a 2-column matrix.
lct (a, n, f)	Returns the n-level local cosine packet table for the data array a using data taper function f.
mother (x, [j, k], s)	Returns the value at x of the mother wavelet packet based on the filter represented by string s at scale j and position k. If j and k are omitted, they are assumed to be 0.
mother2d (x, y, [j, kx, ky], s, ds)	Returns the value at (x,y) of the 2D mother wavelet packet based on the filter represented by string s at scale j and position (kx,ky). String ds is either "H", "V", or "D", and indicates either the horizontal, vertical, or diagonal wavelet. If j, kx, and ky are omitted, they are assumed to be 0.

put_detail (w, l, y)	Returns the result of inserting the vector y into the index-1 block of wavelet packet coefficients of w at level l.
put_detail2d (M, l, N, ds)	Returns the result of inserting the matrix N into the subband indicated by the string ds of wavelet packet coefficients at level l of the 2-dimensional wavelet packet transform M. ds is either "H", "V", or "D", and indicates either the index-(1,0), index-(0,1), or index-(1,1) subband.
put_smooth (w, l, y)	Returns the result of inserting the vector y into the index-0 block of wavelet packet coefficients of w at level l.
put_smooth2d (M, l, N)	Returns the result of inserting the matrix N into the index-(0,0) subband of wavelet packet coefficients at level l of the 2-dimensional wavelet packet transform M.
put_subband (w, l, n, y)	Returns the result of inserting the vector y into the nth block of wavelet packet coefficients at level l of the wavelet packet transform w. w is a vector.
put_subband2d (M, l, m, n, N)	Returns the result of inserting the matrix N into the (m,n)th subband of wavelet packet coefficients at level l of the 2-dimensional wavelet packet transform M. M is an array.
swaveterp (x, v, j, n, f)	Returns the value at x of the symmetric multiresolution approximation of v corresponding to the nth subband at level j, based on the filter f. f is an array returned by bl, Bspline, or daublet(2).
swaveterp2d (x, y, v, j, m, n, f)	Returns the value at (x,y) of the symmetric multiresolution approximation of v corresponding to the (m,n) th subband at level j, based on the filter f. f is an array returned by bl, Bspline, or daublet(2).
symmlet (n)	Returns the low-pass filter of the n-coefficient least asymmetric Daubechies wavelet. n is an even integer between 4 and 20.
wavebs (n, d)	Returns the d-dimensional wavelet packet basis corresponding to the wavelet transform of level n.
wavelet (x, j, k, n, f, [o, [i]])	Returns the value at x of the nth wavelet packet based on filter f at scale j and position k. f is either a vector or a 2-column matrix. f can have an offset o and negation i.
wavelet2d (x, y, j, kx, ky, m, n, f, [o, [i]])	Returns the value at (x,y) of the (m,n)th wavelet packet based on filter f at scale j and position (kx,ky). f is either a vector or a 2-column matrix. f can have an offset o and negation i.
waveterp (x, v, j, n, f, [o, [i]])	Returns the value at x of the periodic multiresolution approximation of v corresponding to the nth subband at level j, based on the filter f. f is either a vector or a 2-column matrix. f can have an offset o and negation i.
waveterp2d (x, y, v, j, m, n, f, [o, [i]])	Returns the value at (x,y) of the periodic multiresolution approximation of v corresponding to the (m,n) th subband at level j, based on the filter f. f is either a vector or a 2-column matrix. f can have an offset o and negation i.
wpt (a, b, f, [o, [i]])	Returns the wavelet packet transform of the array a with respect to basis or level b by the wavelet filter f. b is either an integer or a 2-or-3-column matrix representing a packet basis. f is either a vector or a 2-column matrix. f can have offset o and negation i.

wpti (a, b, f)	Returns the interval wavelet packet transform of the array a with respect to basis or level b by the wavelet filter f. b is either an integer or a 2- or 3-column matrix representing a packet basis. f is a vector.
wpts (a, b, f)	Returns the symmetric wavelet packet transform of the array a with respect to basis or level b by the wavelet filter f. b is either an integer or a 2- or 3-column matrix representing a packet basis. f is either a vector or a 2-column matrix.