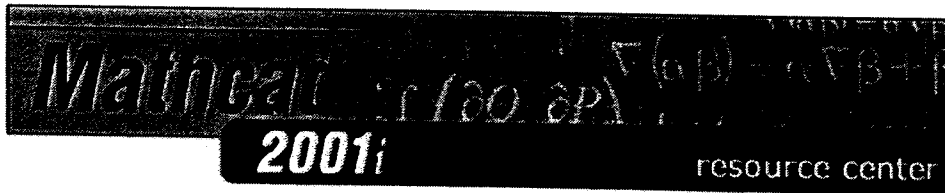










FROM HELP - RESOURCE CENTER
IN MATHCAD



	Collaboratory		Overview and Tutorials
	Web Library		
	Mathcad.com		QuickSheets and Reference Tables
	Support		
	Web Store		Extending Mathcad


New Features

TABLE 3-1

OPERATOR	MATHEMATICAL OPERATION	SOURCE PALETTE	ALTERNATE KEYSTROKE
Square Root	$\sqrt{\quad}$	Calculator toolbar	[] (backslash)
n^{th} Root	$\sqrt[n]{\quad}$	Calculator toolbar	[Ctrl- \backslash]
Absolute Value	$ \quad $	Calculator toolbar	[] (vertical bar)
Factorial	$\quad!$	Calculator toolbar	[!]
Summation	$\sum \quad$	Calculator toolbar	[Shift-4]
Product	$\prod \quad$	Calculus toolbar	[Shift-3]
NOT	$\neg A$	Boolean toolbar	[Ctrl-Shift-1]
AND	$A \wedge B$	Boolean toolbar	[Ctrl-Shift-7]
OR	$A \vee B$	Boolean toolbar	[Ctrl-Shift-6]
XOR	$A \oplus B$	Boolean toolbar	[Ctrl-Shift-5]

Greek letters

There are two ways to enter a Greek variable name in Mathcad:

- Click on the appropriate letter on the Greek toolbar. To see this toolbar, click  on the Math toolbar or choose **Toolbars**⇒**Greek** from the **View** menu.
- Type the *Roman equivalent* of the Greek symbol and then press [Ctrl]G. For example, to enter ϕ , press f [Ctrl]G. See "Greek Letters" on page 475 in the Appendices for a table of Greek letters and their Roman equivalents.

Note Although many uppercase Greek letters look like ordinary capital letters, they are *not* the same. Mathcad distinguishes between Greek and Roman letters, even if they appear the same.

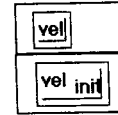
Tip Because it is used so frequently, the Greek letter π can also be typed by pressing [Ctrl][Shift]P.

Literal subscripts


If you include a period in a variable name, Mathcad displays whatever follows the period as a subscript. You can use these *literal subscripts* to create variables with names like vel_{init} and u_{air} .

To create a literal subscript, follow these steps:

1. Type the portion of the name that appears before the subscript.
2. Type a period (.) followed by the portion of the name that is to become the subscript.













Note Do not confuse literal subscripts with *array* subscripts, which are generated with the left bracket

key ([) or by clicking  on the Calculator toolbar. Although they appear similar—a literal subscript appears below the line, like an array subscript, but with a slight space before the subscript—they behave quite differently in computations. A literal subscript is simply a cosmetic part of a variable name. An array subscript represents a reference to an array element. See Chapter 11, “Vectors, Matrices, and Data Arrays,” for a description of how to use subscripts with arrays.

Boolean Operators




Mathcad includes logical or *Boolean* operators on the Boolean toolbar. Unlike other operators, the Boolean operators can return only a zero or a one. Despite this, they can be very useful to perform tests on your expressions.


The following table lists the Boolean operators available on the Boolean toolbar and their meaning. Note that the “Equal to” operator (bold equal sign) is different from the evaluation equal sign you insert by typing =.

Appearance	Button	Description	Keystroke
$w = z$		Equal to; displays as bold equal sign.	[Ctrl] =
$x < y$		Less than	<
$x > y$		Greater than	>
$x \leq y$		Less than or equal to	[Ctrl] 9
$x \geq y$		Greater than or equal to	[Ctrl] 0
$w \neq z$		Not equal to	[Ctrl] 3
$\neg z$		Not	[Ctrl] [Shift] 1
$w \wedge z$		And	[Ctrl] [Shift] 7
$w \vee z$		Or	[Ctrl] [Shift] 6
$w \oplus z$		Xor (Exclusive Or)	[Ctrl] [Shift] 5

Note The Boolean operators return 1 if the expression is true, 0 otherwise. The four operators $>$, $<$, \leq , and \geq cannot take complex numbers because the concepts of greater than and less than lose their meaning in the complex plane.

Tip The comparative boolean operators such as $<$ and $>$ can also be used to compare *strings*. Mathcad compares two strings character by character by determining the ASCII codes of the characters. For example, the string “Euler” precedes the string “Mach” in ASCII order and so the expression (“Euler” $<$ “Mach”) evaluates to 1. To determine the character ordering Mathcad uses in comparing strings, see “ASCII codes” on page 478 in the Appendices.

$A^{(n)}$		Array superscript. n th column of array A . Returns a vector.
v_n		Vector subscript. n th element of a vector.
$A_{m,n}$		Matrix subscript. m , n th element of a matrix.

Tip Operators and functions that expect vectors always expect column vectors. They do not apply to row vectors. To change a row vector into a column vector, use the transpose operator by clicking  on the Matrix toolbar.

Trigonometric Functions

$\text{angle}(x, y)$	Returns the angle (in radians) from the positive x -axis to point (x, y) in the x - y plane. The result is between 0 and 2π .
$\cos(z)$	Returns the cosine of z . In a right triangle, this is the ratio of the length of the side <i>adjacent</i> to the angle over the length of the hypotenuse.
$\cot(z)$	Returns $1/\tan(z)$, the cotangent of z . z should not be a multiple of π .
$\csc(z)$	Returns $1/\sin(z)$, the cosecant of z . z should not be a multiple of π .
$\sec(z)$	Returns $1/\cos(z)$, the secant of z . z should not be an odd multiple of $\pi/2$.
$\sin(z)$	Returns the sine of z . In a right triangle, this is the ratio of the length of the side <i>opposite</i> the angle over the length of the hypotenuse.
$\tan(z)$	Returns $\sin(z)/\cos(z)$, the tangent of z . In a right triangle, this is the ratio of the length of the side <i>opposite</i> the angle over the length of the side <i>adjacent</i> to the angle. z should not be an odd multiple of $\pi/2$.

Mathcad's trig functions and their inverses accept any scalar argument: real, complex, or imaginary. They also return complex numbers wherever appropriate.

Note Trigonometric functions expect their arguments in *radians*. To pass an argument in degrees, use the built-in unit *deg*. For example, to evaluate the sine of 45 degrees, type `sin(45*deg)`.

Inverse Trigonometric Functions

$\text{acos}(z)$	Returns the angle (in radians) whose cosine is z .
$\text{acot}(z)$	Returns the angle (in radians) whose cotangent is z .
$\text{acsc}(z)$	Returns the angle (in radians) whose cosecant is z .
$\text{asec}(z)$	Returns the angle (in radians) whose secant is z .
$\text{asin}(z)$	Returns the angle (in radians) whose sine is z .
$\text{atan}(z)$	Returns the angle (in radians) whose tangent is z .
$\text{atan2}(x, y)$	Returns the angle (in radians) from the positive x -axis to point (x, y) in the x - y plane.

With the exception of *atan2* and *acot*, the inverse trigonometric functions can take either a real or complex argument and return an angle in radians between $-\pi/2$ and $\pi/2$, or the principal value in the case of a complex argument. *atan2* takes only real arguments and returns a result between $-\pi$ and π , *acot* returns an angle in radians between 0 and π for a real argument or the principal value in the case of a complex argument.

To convert a result into degrees, either divide the result by the built-in unit *deg* or type *deg* in the units placeholder

Log and Exponential Functions

$\exp(z)$	Returns e raised to the power z .
$\ln(z)$	Returns the natural log of z . ($z \neq 0$).
$\log(z, b)$	Returns the base b logarithm of z . ($z \neq 0, b \neq 0$). If b is omitted, returns the base 10 logarithm.

Mathcad's exponential and logarithmic functions can accept and return complex arguments. \ln returns the *principal branch* of the natural log function.

Complex Numbers

$\arg(z)$	Returns the angle in complex plane from real axis to z . The result is between $-\pi$ and π radians.
$\text{csgn}(z)$	Returns 0 if $z = 0$, 1 if $\text{Re}(z) > 0$ or ($\text{Re}(z) = 0$ and $\text{Im}(z) > 0$), -1 otherwise.
$\text{Im}(z)$	Returns the imaginary part of a number z .
$\text{Re}(z)$	Returns the real part of a number z .
$\text{signum}(z)$	Returns 1 if $z = 0$, $z/ z $ otherwise.

Piecewise Continuous Functions


$\text{if}(\text{cond}, \text{tvl}, \text{fvl})$	Returns tvl if cond is nonzero (true), fvl if cond is zero (false). cond is usually a Boolean expression.
$\delta(m, n)$	Kronecker's delta function. Returns 1 if $m = n$, 0 otherwise. Both arguments must be integers.
$\epsilon(i, j, k)$	Completely antisymmetric tensor of rank 3. i, j , and k must be integers between 0 and 2 inclusive (or between ORIGIN and ORIGIN + 2 inclusive if ORIGIN $\neq 0$). Result is 0 if any two are the same, 1 for even permutations, -1 for odd permutations.
$\Phi(x)$	Heaviside step function. Returns 1 if $x \geq 0$, 0 otherwise. x must be real.
$\text{sign}(x)$	Returns 0 if $x = 0$, 1 if $x > 0$, and -1 otherwise. x must be real.

Note The *if* function is useful for branching in calculation: choosing one of two values based on a condition. Although the first argument of the *if* function, *cond*, can be any expression at all, it is usually a Boolean expression that relates two math expressions with a Boolean operator.

Truncation and Round-Off Functions

$\text{ceil}(x)$	Least integer $\geq x$ (x real).
$\text{floor}(x)$	Greatest integer $\leq x$ (x real).
$\text{round}(x, n)$	Rounds real number x to n decimal places. If $n < 0$, x is rounded to the left of the decimal point. If n is omitted, returns x rounded to the nearest integer.
$\text{trunc}(x)$	Returns the integer part of a real number x by removing the fractional part.

Vector and Matrix Functions

Note that functions that expect vectors always expect column vectors rather than row vectors. To change a row vector into a column vector, use the transpose operator (click  on the Matrix toolbar).

Size and Scope of an Array

- `cols(A)` Returns the number of columns in array **A**. If **A** is a scalar, returns 0.
- `last(v)` Returns the index of the last element in vector **v**.
- `length(v)` Returns the number of elements in vector **v**.
- `max(A, B, C, ...)` Returns the largest of the strings, arrays, or scalars **A, B, C, ...**. If any value is complex, returns the largest real part plus *i* times the largest imaginary part.
- `min(A, B, C, ...)` Returns the smallest of the strings, arrays, or scalars **A, B, C, ...**. If any value is complex, returns the smallest real part plus *i* times the smallest imaginary part.
- `rows(A)` Returns the number of rows in array **A**. If **A** is a scalar, returns 0.

Special Types of Matrices


- `diag(v)` Returns a diagonal matrix containing on its diagonal the elements of **v**.
- `geninv(A)` Returns the left inverse matrix **L** of **A**, such that $\mathbf{L} \cdot \mathbf{A} = \mathbf{I}$, where **I** is the identity matrix having the same number of columns as **A**. Matrix **A** is an $m \times n$ real-valued matrix, where $m \geq n$.
- `identity(n)` Returns an $n \times n$ matrix of 0's with 1's on the diagonal.
- `rref(A)` Returns the reduced-row echelon form of **A**.

Finding Roots

- `polyroots(v)` Returns the roots of an n th degree polynomial whose coefficients are in **v**, a vector of length $n + 1$. Returns a vector of length n .
- `root(f(z), z)` Returns the value of z which the expression $f(z)$ is equal to 0. The arguments are a real- or complex-valued expression $f(z)$ and a real or complex scalar, z . Must be preceded in the worksheet by a guess value for z . Returns a scalar.
- `root(f(z), z, a, b)` Returns the value of z lying between a and b at which the expression $f(z)$ is equal to 0. The arguments to this function are a real-valued expression $f(z)$, a real scalar, z , and real endpoints $a < b$. No guess value for z is required. Returns a scalar.

Solve Blocks

The general form for using system solving functions in Mathcad is within the body of a *solve block*. There are four general steps to creating a solve block. These are:

1. Provide an initial guess (definition) for each of the unknowns you intend to solve for. Mathcad solves equations by making iterative calculations that ultimately converge on a valid solution. The initial guesses you provide give Mathcad a place to start searching for solutions. Guess values are usually required for all systems.
2. Type the word *Given* in a separate math region below the guess definitions. This tells Mathcad that what follows is a system of constraint equations. Be sure you don't type "Given" in a text region.
3. Now enter the constraints (equalities and inequalities) in any order below the word *Given*. Make sure you use the bold equal symbol (click  on the Boolean toolbar or press [Ctrl]=) for any equality. You can separate the left and right sides of an inequality with any of the symbols <, >, ≤, and ≥.
4. Enter any equation that involves one of the functions *Find*, *Maximize*, *Minimize*, or *Minerr* below the constraints.

Note Solve blocks cannot be nested inside each other—each solve block can have only one *Given* and one *Find* (or *Maximize*, *Minimize*, or *Minerr*). You can, however, define a function like $f(x) := \text{Find}(x)$ at the end of one solve block and refer to this function in another solve block.

Solve Blocks

$\text{Find}(z_0, z_1, \dots)$ Returns values of z_0, z_1, \dots that satisfy the constraints in a solve block. z_0, z_1, \dots are real or complex scalars, vectors, arrays, or individual elements of vectors equal in number to the number of unknowns in the system. Returns a scalar for a single unknown; otherwise returns a vector of solutions.

$\text{Maximize}(f, z_0, z_1, \dots)$ Returns values of z_0, z_1, \dots that make the function f take on its largest value. z_0, z_1, \dots are real or complex scalars, vectors, arrays, or individual elements of vectors equal in number to the number of unknowns in the system. Returns a scalar for a single unknown; otherwise returns a vector of solutions. Solve block constraints are optional.

$\text{Minerr}(z_0, z_1, \dots)$ Returns values of z_0, z_1, \dots that come closest to satisfying the constraints in a solve block. z_0, z_1, \dots are real or complex scalars, vectors, arrays, or individual elements of vectors equal in number to the number of unknowns in the system. Returns a scalar for a single unknown; otherwise returns a vector of solutions.

$\text{Minimize}(f, z_0, z_1, \dots)$ Returns values of z_0, z_1, \dots that make the function f take on its smallest value. z_0, z_1, \dots are real or complex scalars, vectors, arrays, or individual elements of vectors equal in number to the number of unknowns in the system. Returns a scalar for a single unknown; otherwise returns a vector of solutions. Solve block constraints are optional.

Tip Unlike most Mathcad functions, the solving functions *Find*, *Maximize*, *Minerr*, and *Minimize* can be entered in math regions with either an initial lowercase or an initial capital letter.

SI Units

Base Units

m (meter), *length*
A (ampere), *current*
mole or mol, *substance*.

kg (kilogram), *mass*
K (kelvin), *temperature*

s (second), *time*
cd (candela), *luminosity*

Angular Measure

$$\text{rad} = 1$$

$$\text{deg} = \frac{\pi}{180} \cdot \text{rad}$$

$$\text{sr} = 1 \cdot \text{sr}$$

Length

cm = 0.01 · m
ft = 0.3048 · m
mi = 5280 · ft

km = 1000 · m
in = 2.54 · cm

mm = 0.001 · m
yd = 3 · ft

Mass

gm = 10^{-3} · kg
mg = 10^{-3} · gm
oz = $\frac{\text{lb}}{16}$

tonne = 1000 · kg
ton = 2000 · lb

lb = 453.59237 · gm
slug = 32.174 · lb

Time

min = 60 · s
yr = 365.2422 · day

$$\text{hr} = 3600 \cdot \text{s}$$

$$\text{day} = 24 \cdot \text{hr}$$

Area, Volume

hectare = 10^4 · m²
mL = 10^{-3} · L

acre = 4840 · yd²
fl_oz = 29.57353 · cm³

L = 0.001 · m³
gal = 128 · fl_oz

Velocity, Acceleration

$$\text{mph} = \frac{\text{mi}}{\text{hr}}$$

$$\text{kph} = \frac{\text{km}}{\text{hr}}$$

$$g = 9.80665 \cdot \frac{\text{m}}{\text{s}^2}$$

Force, Energy, Power

$$\text{N} = \text{kg} \cdot \frac{\text{m}}{\text{s}^2}$$

$$\text{dyne} = 10^{-5} \cdot \text{N}$$

$$\text{lbf} = \text{g} \cdot \text{lb}$$

$$\text{kgf} = \text{g} \cdot \text{kg}$$

$$\text{J} = \text{N} \cdot \text{m}$$

$$\text{erg} = 10^{-7} \cdot \text{J}$$

$$\text{cal} = 4.1868 \cdot \text{J}$$

$$\text{kcal} = 1000 \cdot \text{cal}$$

$$\text{BTU} = 1.05506 \cdot 10^3 \cdot \text{J}$$

$$\text{W} = \frac{\text{J}}{\text{s}}$$

$$\text{kW} = 1000 \cdot \text{W}$$

$$\text{hp} = 550 \cdot \frac{\text{ft} \cdot \text{lbf}}{\text{s}}$$

U.S. Customary Units

Base Units

ft (foot), *length*
coul (coulomb), *charge*

lb (pound), *mass*
K (kelvin), *temperature*

sec (second), *time*

Angular Measure

rad = 1

deg = $\frac{\pi}{180}$ · rad

Length

in = $\frac{\text{ft}}{12}$

m = $\frac{\text{ft}}{0.3048}$

yd = 3 · ft

cm = 0.01 · m

mi = 5280 · ft

km = 1000 · m

mm = 0.001 · m

Mass

slug = 32.174 · lb

oz = $\frac{\text{lb}}{16}$

ton = 2000 · lb

kg = $\frac{\text{lb}}{0.45359237}$

tonne = 1000 · kg

gm = 10^{-3} · kg

mg = 10^{-3} · gm

Time

min = 60 · sec

hr = 3600 · sec

day = 24 · hr

yr = 365.2422 · day

Area, Volume

acre = 4840 · yd²

hectare = 10⁴ · m²

fl_oz = 29.57353 · cm³

liter = 0.035 · ft³

mL = 10⁻³ · liter

gal = 128 · fl_oz

Velocity, Acceleration

mph = $\frac{\text{mi}}{\text{hr}}$

kph = $\frac{\text{km}}{\text{hr}}$

g = 32.174 · $\frac{\text{ft}}{\text{sec}^2}$

Force, Energy, Power

lbf = g · lb

newton = kg · $\frac{\text{m}}{\text{sec}^2}$

dyne = 10⁻⁵ · newton

kgf = g · kg

joule = newton · m

erg = 10⁻⁷ · joule

cal = 4.1868 · joule

kcal = 1000 · cal

BTU = 1.05506 · 10³ · joule

watt = $\frac{\text{joule}}{\text{sec}}$

hp = 550 · $\frac{\text{ft} \cdot \text{lbf}}{\text{sec}}$

kW = 1000 · watt