



A Windows Calculation Facility

© 1993,1994 Davut Z. Korkut

Introduction

[What is Zcalc?](#)

[Limitations.](#)

[How do I Register Zcalc?](#)

Variables and Equations

[The Algebraic Engine](#) [Constants](#)

[Operators](#) [Variables](#)

[Equations](#) [Numeric Options](#)

Zcalc Modes

[Full Calculator](#) [Line Calculator](#)

[What if Calculator](#) [Tables & Graphs](#)

Reference

[Function Reference](#) [Using Zcalc.ini](#)

What is Zcalc?

The Z in Zcalc comes from the traditional notation of representing general imaginary numbers as the variable z . Zcalc is a multi-purpose general calculator that has the following features:

- Based on algebraic system of entry. Expressions are entered in natural algebraic notation with normal operational precedence.
- Variables and equations can have long alpha-numeric names.
- Supports *real* and *imaginary* numbers. Imaginary numbers can be expressed in polar and rectangular form.
- Using the clipboard, values from other applications can be stored into Zcalc variables.
- A full range of built in functions are available.
- Four modes of operation: A full featured tape calculator, a small sized line calculator, a table and graph calculator and a "what-if" calculator.
- The full calculator is tape-based push button calculator that can be fully operated by the mouse. Four user defined strings are available in the full calculator mode. All functions and operations are available with out requiring any shift keys. The tape saves the results of all operations. Tape entries may be edited, notated and re-executed. The tape may be saved in ASCII form for reference.
- The line calculator appears in a smaller window in the lower right-hand corner of the screen. It has one text field where operations can be executed and results viewed. It keeps a history of previous calculations and is intended to be used in conjunction with other applications.
- The table and graph calculator allows the calculation of tables and graphs based on existing equations and variables. Up to 1000 data points can be used.
- The what-if calculator is used to view a list of variables and equations, then modify the variables to see the effect on the equations.

Limitations

Zcalc is not a spread sheet nor a high level symbolic calculator available in some expensive packages. Its graphing capability is simple and basic. It is a utility to be used for quick and easy calculations in support of other applications.

Zcalc has the following limitations.

- Variable and equation names are limited to 40 characters.
- The number of variables and equations that may be stored is dependent only on memory. However, the higher the number, the slower the response.
- Only four user defined strings are supported in this version.
- The number of data points in a table or graph is limited to 1000.
- Expressions can be rather long and complicated but not infinitely so. Zcalc may complain of a stack overflow if it cannot evaluate an expression.

Disclaimer

The author has attempted to provide a well-behaved bug-free application. However, as with any large complicated program, the application cannot be guaranteed to be free from bugs. The author will attempt to correct any bugs found. However, the user must determine the applicability of the program and accepts responsibility for its use. The author cannot be held liable for any damages, real, imagined or consequential, resulting from the use of this program.

Registration

Zcalc is not free nor public domain. If you plan to use it then please register it! In the spirit of true shareware, Zcalc has not been limited or crippled in any way. Help me keep developing it into a tool that can help you.

Zcalc may be registered by sending \$39.95 in check or money order to:

Davut Z. Korkut
451 Evergreen Ct.
Slidell, LA 70458

THANKS for the support!

The Algebraic Engine

Zcalc has an embedded calculator engine that is based on the algebraic system of entry. Data is stored in variables. Equations are based on [expressions](#) that consist of variable names, constants and other equation names. Equation expressions are entered in a natural left to right algebraic form. For example, the following is a valid Zcalc equation:

TheValue = 50 * 30 ^ (a + b) * sin(theta)

The equation name is *TheValue*. *a*, *b*, and *theta* could be previously defined equations or variables. *sin* is a built in trigonometric function. ***, *^*, and *+* are operators. The parentheses override normal operator precedence.

Equations are parsed and stored in an internal format. However, Zcalc stores the exact way the equation was typed so you can use case and spaces for formatting.

Expressions

A Zcalc *expression* consists of constants, variables names, equation names, operators, parentheses and optional spaces.

Some examples of valid expressions are

5 + 3 * (1E-1)

[5:5] + [4@1.5r] + radius

((1+j*5) * volume) * density

pi * radius ^ 2

Constants

Real Constants

Constants are non-varying numeric values that can be *real* or *imaginary*. A REAL constant is simply a string of numeric digits as follows. Note that [scientific notation](#) is supported:

5.0 3.4 3 10.4E25 5e-3

Imaginary Constants

IMAGINARY constants are imaginary numbers that are expressed in one of the following forms:

[x : y]

[r @ theta]

(x + j * y)

The first two forms REQUIRE brackets ("[]") to surround the values. The first form is rectangular notation where *x* is the real portion of the number and *y* is the imaginary portion. the second form uses polar notation where *r* is the magnitude and *theta* is the angle. *Theta* may be followed by the letter *r* to denote radians or *d* to denote degrees or nothing to use the current default. The last form is the [expression](#) form where the number is the sum of the *real* plus the *imaginary* parts. The imaginary part is preceded by the factor "j" where j is the square root of negative 1. The last form actually invokes the algebraic expression evaluator. The following are samples of imaginary constants.

[5:3] [-1:4E-2] [1@45d] [2@1.54r] [3@1.2] (3 + j*3)

It should be noted that ALL numbers are stored internally as *imaginary* numbers. A *real* number is merely an imaginary number with a 0 value imaginary part.

When Requested for a Value

Many times Zcalc will request a value in a text box. You can, of course enter any valid *real* or *imaginary* constant, but you may also enter any valid [expression](#). The expression will be evaluated into a constant for you. Please note that the expression is NOT saved, only the VALUE of the expression. If you wish to store expressions, you must use [equations](#).

Scientific Notation

Scientific notation allows very large numbers to be expressed in a short form. The number is expressed as a mantissa, or the significant digits of the number, and the exponent, or the power of 10 to which the mantissa must be raised in order to represent the number. An example of scientific notation is:

1.245e3 = 1,245 where 1.245 is the mantissa and 3 is the exponent

Operators

Operators separate constants, variable names and equation names. All operators are binary except the "-" operator when it appears at the beginning of an expression. The following is a list of available operators:

- 1) ^ exponentiate
- 2) * multiply
- 3) / divide
- 4) + add
- 5) - subtract or negate

Operator Precedence

Operators have the following PRECEDENCE. Operations that have higher precedence are performed first. Operators with the same precedence are performed left to right. Parentheses can be used to override precedence. Operator precedence is as follows:

(negate)	exponentiate	multiply	divide	add	subtract
4	3	2	2	1	1

For example:

1 + 2 * 3 = 7 but (1 + 2) * 3 = 9
-3 ^ 2 = -9 but (- 3) ^ 2 = 9

Equations

Equations are named objects that store an expression that evaluates to a value. An equation has the following stored attributes:

A name, a character string [expression](#), the current value expressed as a [constant](#), and an optional [description](#)

How to Create or Edit an Equation

From the main window MENU choose:

Define .. Equations

From the FULL CALCULATOR mode choose

Eqns .. Edit Equations

From the WHAT IF mode MENU choose

Equations

In the TABLES AND GRAPHS mode:

Press the EQUATIONS command button.

At this point you will be presented with the [Equation Edit Dialog](#) where you can Add, Edit and Delete variables.

In the LINE CALCULATOR mode, equations can easily be defined via the input text box as follows:

Enter the name of the equation followed by := (colon equal sign) followed by the expression. If the equation exists, the new expression will replace the old expression. For example:

volume := a * pi * radius ^ 2

will create the equation *volume* if it does not exist and assign the expression $a * \pi * radius^2$ to it

Names

Names are user defined entities associated with [variables](#) and [equations](#). The rules for names are as follows:

- 1 The entire name must be a contiguous string of the any combinations of the characters A - Z, 0 - 9, and "_" (underscore).
- 2 The entire name cannot exceed 40 characters.
- 3 The first letter must alphabetic, i.e. A - Z.
- 4 The name is case-insensitive.
- 5 The name must not be the name of a built-in function.

Descriptions

Descriptions are text fields associated with a [variable](#) or [equation](#). They are totally optional. Carriage returns may be inserted by using the CTRL-ENTER key combination.

Variables

Variables are named objects that store a value that may be changed at any time. A variable has the following stored attributes:

A name, the current value expressed as a constant, and an optional description.

How to Create or Edit a Variable

From the main window MENU choose

Define -> Variables

From the FULL CALCULATOR mode MENU choose

Vars -> Edit Variables

From the WHAT IF mode MENU choose

Variables

In the TABLES AND GRAPHS mode click on the *Variables* command button. At this point you will be presented with the Variable Edit Dialog, where you can add, edit and delete variables

In the LINE CALCULATOR mode, variables can be easily edited via the input text box as follows.

Enter the name of the variable followed by the equal sign "=" and press "ENTER"
For example entering

alpha = 3.5

will create the variable *alpha* if it does not exist. If *alpha* exists, the value "3.5" will be assigned to it.

To delete a variable, enter the name followed by the equal sign and NO value. For example,

alpha =

will delete the variable *alpha*.

Variable Edit Dialog

The variable edit dialog allows you to maintain your collection of variables. You must enter a [name](#), the current value expressed as a [constant](#), and an optional [description](#).

To ADD a variable:

Enter the variable name then click on the *Add* command button.

To UPDATE the [value](#) of an existing variable:

Choose an existing name using the name combo box list or type in the name of an existing variable. The current value and description will be displayed. Make the required changes and click on the *Update* command button. **IMPORTANT:** If you do not click on the *Update* button, your changes will not be stored.

To DELETE a variable:

Choose an existing name using the name combo box list or type in the name of an existing variable. The current value and description will be displayed. Click on the *Delete* command button.

Note that values can be entered in as any valid [constant](#) or [expression](#).

To recall a list of previously used expressions, press the *Recall* command button. To change how constants are displayed, press the [Options](#) command button.

When using an expression to store a value to a variable, you may build the expression using lists of currently defined variables, expressions and functions. You merely point and click on the names you want inserted. Just remember, the expression is evaluated to a constant prior to being stored. Place the caret in the *value* field and then:

To insert a Variable name

Click on the *Variables* command button and you will be presented with a grid of existing variables. Double-click on the name you want inserted.

To insert an Equation name

Click on the *Equations* command button and you will be presented with a grid of existing equations. Double-click on the name you want inserted.

To insert a Built-in Function name

Click on the *Functions* command button and you will be presented with a grid of existing built-in functions. Double-click on the function you want inserted.

Once you have edited your collection of variables you may wish to [save](#) them

Equation Edit Dialog

The equation edit dialog allows you to maintain your collection of equations.

To ADD an equation, enter the

A name, a character string expression, the current value expressed as a constant, and an optional description

Then click on the *Add* command button. The current value of the expression will be displayed in the value text box.

To UPDATE the value of an existing equation:

Choose an existing name using the name combo box list or type in the name of an existing equation. The current expression, value and description will be displayed. Make the required changes and click on the *Update* command button. **IMPORTANT:** If you do not click on the *Update* button, your changes will not be stored.

To DELETE an equation.

Choose an existing equation using the name combo box list or type in the name of an existing equation. The current expression, value and description will be displayed. Click on the *Delete* command button.

To control how values are displayed, press the Options command button.

To calculate the current VALUE of an expression without updating it press the *Calculate* command button.

To edit your current collection of variables press the Create Vars command button.

When entering the expression string you may choose variables, equations and function names from pick lists by choosing the appropriate command button. By double clicking on the desired name in the list, the name will be inserted into the expression text box.

IMPORTANT: DO NOT PRESS "Ctrl-Enter" or "Ctrl-Tab to insert a carriage returns or tabs into the expression box. You may use spaces freely, but carriage returns and tabs will cause a SYNTAX error.

After entering your set of equations, you may wish to save them

Numeric Options

There are several options that affect how numbers are displayed and interpreted.

Imaginary Number Format

You can choose how imaginary numbers are displayed.

Rectangular format:

[x : y]

Polar format:

[r @ theta]

Expression format

(x + j * y)

Click on the corresponding option button.

Virtual Zero

There may be times when you wish to display very small numbers as a zero. Any number smaller than virtual zero will be output as zero. *Notice that the number is not set to zero, only displayed as zero..*

Precision

This setting represents the number of significant digits with which a result will be displayed.

Default Angle Measurement

When an number that represents an angular measurement is displayed it will be displayed in *radians* or *degrees* as per the setting of this option. Also, any number input that is an angle will be assumed to be measured according to this option. For example:

[3 @ 45]

will be interpreted as 3 at an angle of 45 degrees if the degrees options is chosen. However, when entering angles in imaginary numbers of polar format you can override the default by placing an r or d after the angel portion of the number. Therefore

[3 @ 45r]

will be interpreted as 3 @ angle of 45 *radians*, regardless of the setting of the default angle measurement.

Also, be careful when using trigonometric functions to make note of the default angle measurement. For example the two expressions:

sin(45) and arctan(0.9)

will yield different results depending on the default measurement setting.

Full Calculator Mode

The full calculator mode emulates a full sized desk calculator with tape output. The calculator is initiated in the center of the screen. It is intended be fully controlled by a mouse. However, the calculator result text box can accept keyboard input as in the [Line Calculator](#).

Points of Interest

- Expression components are inserted at current caret position in the result text box by clicking on the corresponding component keys.
- .Alpha-numeric characters can be entered by clicking on the *Alpha* key and continuing to click on the desired characters. Clicking on *OK* will insert the characters entered to the current caret position of the result text box.
- By turning the tape output on (clicking on the on option button), any expression that is entered in the text box is appended onto the end of the tape. You can create annotations by appending a (single quote) to the end of an expression and then typing the comment. The expression is not evaluated after the single quote. The single quote can also be entered at the current caret position by clicking on the *Note* key.
- You can edit any line on the tape by clicking on the line. The edited line can then be copied to the result text box or back to the tape.
- .Highlighted lines in the tape can either be cut/copied to the clipboard, saved to a file, or printed. *Saving* and *Loading* tapes from the File menu is another way of storing equations and variable values.
- The current character can be erased by clicking on the *Bcksp* key. The result text box can be erased by clicking on the *Entry* key. The tape can be erased by clicking on the *Tape* key.
- You can cut and/or copy any highlighted text in the result text box. Using this method you can store numeric values from other applications into Zcalc variables.
- You can erase the entire input buffer by pressing *ctrl-backspace*.
- [Display Options](#) can be changed.
- Variables and Equations can be defined either by entering the = and := operators into the result text box to create and change [variables](#) and [equations](#) or using the Menu selections described below.
- Variable and Equations references (names) can be inserted at the current caret position in the result text box by choosing the Vars/List Variables or Eqns/List Equations menu selections respectively and then clicking on the desired item. Variables and Equations can be also be *created* though the Vars/Create Variables and Eqns/Create Equations
- You may define up to 4 [user defined strings](#) through the Options/User Defined String menu option.

Line Calculator Mode

The line calculator mode is a small window calculator. The calculator is initiated in the lower right hand corner of the screen out of the way of other windows. It is intended be used to easily augment other applications. Expressions can only be entered directly into the result text box via the keyboard.

Points of Interest

- You can cut and/or copy any highlighted text in the result text box. Using this method you can store numeric values from other applications into Zcalc variables.
- Variables and Equations can be defined either by using the = and := operators to create and change [variables](#) and [equations](#).
- You can erase the entire input buffer by pressing *ctrl-backspace*.
- Variable, Equations references (names) and Function names can be inserted at the current caret position in the result text box by choosing the Vars, Eqns, or Fnctns menu selections respectively and then clicking on the desired item.

What If Mode

The What If mode allows you to list the [variables](#) and [equations](#) to see how a group of equations is influenced by changes in a group of variables. The screen is divided into an input and output list. Using the combo boxes at the bottom of each list, you can choose which variables and equations you want to consider. Once the variable list is established, you can modify individual values by clicking on the line where the variable name is displayed. The current value is displayed in the text box at the bottom of the variable list. You can then input the value that you want. The affected equations will be automatically updated in the equation list. You can use the menu selections *Variables* and *Equations* to bring up the [Equation](#) and [Variable](#) dialogs to create and edit variables and equations. If, at any time, the equation list does not reflect current variable settings, the *Calculate* command button will be enabled. Click on this button to update the equation results. Display [options](#) can be modified by choosing the *Options* menu selection.

Table and Graph Mode

The Table and Graph mode allows you to create tables and graphs of an [equation](#) versus a [variable](#). You must fill in the following items prior to clicking on the *Calculate* button:

Equation: Type in the name of the equation you wish to use or click on the select arrow of the combo box to display a list of currently defined equations.

Variable: Type in the name of the variable you wish to use or click on the select arrow of the combo box to display a list of currently defined variables.

Start: Type in the starting [value](#) of the variable.

End: Type in the ending [value](#) of the variable.

Increment: Type in the [value](#) of the increment to be used in calculations.

Type of Increment: Type in the type of increment to use. If you choose *By Size* then the Increment will be assumed to be an actual value so that each calculation point will be the previous value PLUS the increment. For example if you choose to start at 1 and end at 10 with a increment of 1 *by size*, the variable will be set to 1, 2, 3...10 as calculations are performed. If you choose *By Number of Points*, the increment will be interpreted to mean that you wish to have a fixed number of points. The actual value used in calculations will be $(\text{end} - \text{start})/\text{increment}$. For example, if you choose 0 as the start and 20 with an increment of 5 by *number of points*, the variable will be set to 0, 4,10,15 and 20 as calculations are performed.

To actually perform the calculations, click on the *Calculate* button. This button will become disabled when a valid set of values have been calculated. After calculation the Table and Graph buttons will become enabled. Choose [Graph](#) to view a graph of the equations and [Table](#) to view a table of values.

You can click on the *Variables* button to display the [Variable Edit Dialog](#), the Equations button to display the [Equation Edit Dialog](#) and the Options to display the [Options Dialog](#).

Graph Form

The graph form displays the set of current [equation](#) values versus the current [variable](#) values in graph form according to the [graph options](#).

- .You can set graph options by clicking on the *Options* command button.
- You can send the graph to the printer by clicking on the *Print* command button.
- You can copy the current graph to the clipboard by clicking the *Copy* command button.

Graph Options

You can control how a graph will appear from the Graph Options Dialog.

- Orientation:** You can graph the [equation](#) vs. [variable](#) or variable vs. equation by clicking the corresponding option.
- Grid Marks:** You can turn on horizontal grids and/or vertical grids by clicking the corresponding check box.
- Graph Type:** You can use a linear scale, a log/linear scale or a polar scale by clicking the corresponding option.
- Mark Data Points:** Check this box if you want actual calculated points to be marked.
- Thick Lines:** Check this box if you want the graph line to be thickened.
- Stats:** Various statistics can be shown on the graph by choosing the desired statistic from this check box. For instance, you can show the average value by choosing the *Mean* statistic.
- Data Type:** You can choose which part of the equation values is plotted by choosing the corresponding Data Type. For example, a regular x-y graph will be plotted if you choose *Re(f(x)) vs. x*.
- Graph Title:** Enter in the optional name of the graph here.
- Variable Label:** Enter in the optional label for the x axis.
- Equation Label:** Enter in the optional label for the y axis.

Table Form

The table form displays the set of current [equation](#) values versus the current [variable](#) values in table format.

- .You can set what columns you wish to display clicking on the Columns check boxes. You can display the variable value (x), the real part of the equation ($\text{Re}(f(x))$), the imaginary part of the equation ($\text{Im}(f(x))$), the Magnitude of the equation ($\text{Mag}(f(x))$) and/or the angle or argument of the equation ($\text{Angle}(f(x))$). Note that you need to click the *Refresh* command button after changing columns to be displayed.
- You can send the table to the printer by clicking on the *Print* command button.
- You can save the current table in ASCII delimited form by clicking the *Save* command button.

User Defined Strings

When using the [Full Calculator Mode](#), you may define up to four strings that may be accessed by command buttons on the calculator form. On this form you enter *names* and associated *strings*. The name you assign will appear on the corresponding button on the calculator form. The default button names are User 1 through User 4

To assign a hot key, in the button name text box, place an ampersand (&) immediately in front of the letter that will be used as the {hot key}. Assign the string that you want passed when the button is clicked or **Alt{hot-key}** is pressed.

Note that these strings are only a macro type substitutions and must represent all or part of a valid Zcalc [expression](#).

Using Zcalc.ini

The file **zcalc.ini** can be used to store default [variables](#), [equations](#) and built in [constants](#). There are four optional sections for the file: **[Variables]**, **[Equations]**, **[Builtins]**, and **[Defaults]**. For the first three sections, each line is of the form `<Name>=<Expression>;<Optional Comment>`. For the [Defaults] section there are four possible lines each of the form `<Name>=<Setting>`, each corresponding to setting in the [Options](#) dialog. See the sample below on how to set up a typical zcalc.ini file. Please note that the file must be placed in the WINDOWS directory. Also note that there are two built in constants that you do not need to define: **PI = 3.14159... and J=SQR(-1)**

Sample File:

[Variables]

```
X=5;Variable X
Y=X+5;Variable Y
Z=5*(3+j*4)*[3@45d];
radius=1;Radius of circle
```

[Equations]

```
Total=x+y+r+z
area=pi*radius^2;area of circle
```

[Builtins]

```
i=j
```

[Defaults]

```
;AngleMeasurement must be Radians or Degrees
AngleMeasurement=Radians
;Precision must be between 0 and 15 inclusive
Precision=5
VirtualZero=1E-100
;Format can be Rectangular, Angular or XplusjY
Format=Rectangular
```

Function Reference

The following built-in functions are supported by Zcalc:

<u>ABS</u>	<u>ANGLE</u>	<u>ARCCOS</u>
<u>ARCCOSH</u>	<u>ARCSIN</u>	<u>ARCSINH</u>
<u>ARCTAN</u>	<u>ARCTAN2</u>	<u>ARCTANH</u>
<u>COS</u>	<u>COSH</u>	<u>EXP</u>
<u>FACTORIAL</u>	<u>IF</u>	<u>IM</u>
<u>LN</u>	<u>LOG</u>	<u>MAX</u>
<u>MIN</u>	<u>RANDOM</u>	<u>RE</u>
<u>SIGN</u>	<u>SIN</u>	<u>SINH</u>
<u>SQR</u>	<u>TAN</u>	<u>TANH</u>
<u>TODEGREES</u>	<u>TORADIANS</u>	

ABS(z)

Returns the *absolute value* of the argument. For *real* arguments, the positive value of the number is returned. For *imaginary* arguments, the *magnitude* of the number is returned.

Examples:

if $x = -5$ then $abs(x) = 5$

if $z = [3:4]$ then $abs(z) = 5$

ANGLE(z)

Returns the *angle* of the argument. The return value depends on the current default angle measurement [option](#).

Examples:

if $x = 5$ then $angle(x) = 0$

if $z = [5@45d]$ then $angle(z) = 45$ if degrees are the default angle measurement.

ARCCOS(z)

Returns the *arccos* of the argument, or the angle whose cos is z. The return value depends on the current default angle measurement [option](#). You can use the [TODEGREES](#) or the [TORADIANS](#) function to override the angle measurement.

Examples:

if $x = 1$ then $\text{arccos}(x) = 0$

if $z = (3 + j * 2)$ then $\text{arccos}(z) = [0.60614:-1.96864]$ if radians are the default angle measurement.

ARCCOSH(z)

Returns the *hyperbolic arccos* of the argument, or the angle whose hyperbolic cos is z. The return value depends on the current default angle measurement [option](#). You can use the [TODEGREES](#) or the [TORADIANS](#) function to override the angle measurement.

Examples:

if $x = 1$ then $\text{arccosh}(x) = 0$

if $z = [1:1]$ then $\text{arccoh}(z) = [1.06128:0.90456]$ if degrees are the default angle measurement.

ARCSIN(z)

Returns the *arcsin* of the argument, or the angle whose sin is z. The return value depends on the current default angle measurement [option](#). You can use the [TODEGREES](#) or the [TORADIANS](#) function to override the angle measurement.

Examples:

if $x = 0$ then $\text{arcsin}(x) = 0$

if $z = (3 + j * 2)$ then $\text{arcsin}(z) = [0.96466:-1.96864]$ if radians are the default angle measurement.

ARCSINH(z)

Returns the *hyperbolic arcsin* of the argument, or the angle whose hyperbolic sin is z. The return value depends on the current default angle measurement [option](#). You can use the [TODEGREES](#) or the [TORADIANS](#) function to override the angle measurement.

Examples:

if x = 0 then *arcsinh*(x) = 0

if z = [1@1r] then *arcsinh*(z) = [0.00021:0.00023] if degrees are the default angle measurement.

ARCTAN(z)

Returns the *arctan* of the argument, or the angle whose tan is z. The return value depends on the current default angle measurement [option](#), and is quadrant independent. You can use the [TODEGREES](#) or the [TORADIANS](#) function to override the angle measurement.

Examples:

if $x = 0$ then $\text{arctan}(x) = 0$

if $x = -1/-1$ then $\text{arctan}(x) = 0.78540$

if $x = 1/1$ then $\text{arctan}(x) = 0.78540$ (notice that the quadrant is ignored)

if $z = (3 + j * 2)$ then $\text{arctan}(z) = [0.02337:0.00256]$ if degrees are the default angle measurement.

ARCTAN2(z1,z2)

Returns the *arctan* of the argument, or the angle whose tan is $\text{Re}(z1)/\text{Re}(z2)$, considering the correct quadrant of the arguments. The return value depends on the current default angle measurement [option](#). You can use the [TODEGREES](#) or the [TORADIANS](#) function to override the angle measurement.

Examples:

if $x = -1/-1$ then $\text{arctan2}(x) = -135$ if degrees are the default angle measurement.

if $x = 1/1$ then $\text{arctan2}(x) = 45$ if degrees are the default angle (Notice that the quadrant is considered.)

if $z1 = [1:1]$ and $z2 = [1:2]$ then $\text{arctan2}(z1,z2) = 45$ if degrees are the default angle measurement. (Notice that only the REAL parts of $z1$ and $z2$ are considered.)

ARCTANH(z)

Returns the *hyperbolic arctan* of the argument, or the angle whose hyperbolic tan is z. The return value depends on the current default angle measurement [option](#). You can use the [TODEGREES](#) or the [TORADIANS](#) function to override the angle measurement.

Examples:

if $x = 0$ then $\operatorname{arctanh}(x) = 0$

if $z = [1@2r]$ then $\operatorname{arctan}(z) = [0.00007:0.00024]$ if degrees are the default angle measurement.

COS(z)

Returns *cos* of the argument. The return value depends on the current default angle measurement [option](#). You can use the [TODEGREES](#) or the [TORADIANS](#) function to convert the argument to the desired angle measurement.

Examples:

if $x = 90$ then $\cos(x) = 0$ if degrees are the default angle measurement.

if $z = (3 + j * 2)$ then $\cos(z) = [-3.72455:-0.51182]$ if radians are the default angle measurement.

COSH(z)

Returns *hyperbolic cos* of the argument.

Examples:

if $x = 0$ then $\cosh(x) = 1$

if $z = (3 + j * 2)$ then $\cosh(z) = [-4.189626:9.109228]$

EXP(z)

Returns the *exponential* of the argument or the constant e raised to the z power.

Examples:

if $x = 1$ then $\exp(1) = 2.71828$

if $z = [3:4]$ then $\exp(z) = [-13.12878:-15.20078]$

FACTORIAL(z)

Returns the *factorial* of the *integer* part of the *magnitude* of the argument or z!.

Examples:

if x = 5 then factorial(x) = 120

if z = [3:4] then factorial(z) = 120

IF(z1,z2,z3,z4)

If $\text{Re}(z1) < 0$, then IF returns $z2$, if $\text{Re}(z1) = 0$ then IF returns $z3$, if $\text{Re}(z1) > 0$ then if returns $z3$

Examples:

if x = 5 then *if*(x,1,2,3) = 3

if z = [0:4] then *if*(z,1,2,3) = 2

IM(z)

Returns the *imaginary part* of the argument.

Examples:

if x = 1 then *im*(1) = 0

if z = [4@90d] then *im*(z) = 4

LN(z)

Returns the *natural logarithm* of the *real* part of the argument.

Examples:

if $x = 0$ then $\ln(x)$ causes an error.

if $z = [3:4]$ then $\ln(z) = [1.60944:0.92730]$

LOG(z)

Returns the *common logarithm* of the *real* part of the argument.

Examples:

if $x = 1$ then $\log(x) = 0$

if $z = [0:4]$ then $\log(z) =$ causes an error.

MAX(z1,z2)

Returns the *larger* of the *real* parts of the two arguments.

Examples:

if $x_1 = 0$ and $x_2 = 3$ then $\max(x_1, x_2) = 3$

if $z = [-3:4]$ then $\max(z, -1) = -1$

MIN(*z*₁,*z*₂)

Returns the smaller of the *real* parts of the two arguments.

Examples:

if $x_1 = 0$ and $x_2 = 3$ then $\min(x_1, x_2) = 0$

if $z = [-3:4]$ then $\min(z, -1) = -3$

RANDOM(z)

Returns random number between 0 and the ABS(z).

Examples:

if x = 1 then *random(x)* = 0.92289

if z = (3+j*4) then *random(z)* = 4.01851

RE(z)

Returns the *real part* of the argument.

Examples:

if $x = 1$ then $re(x) = 1$

if $z = [4@90d]$ then $re(z) = 0$

angle measurement.

SIGN(z)

Returns -1 if $\text{Re}(z) < 0$, 0 if $\text{Re}(z) = 0$ and 1 if $\text{Re}(z) > 0$

Examples:

if $x = -1$ then $\text{sign}(x) = -1$

if $z = [0:4]$ then $\text{sign}(z) = 0$

SIN(z)

Returns *sin* of the argument. The return value depends on the current default angle measurement [option](#). You can use the [TODEGREES](#) or the [TORADIANS](#) function to convert the argument to the desired angle measurement.

Examples:

if $x = 90$ then $\sin(x) = 1$ if degrees are the default angle measurement.

if $z = (3 + j * 2)$ then $\sin(z) = [0.530921:-3.590565]$ if radians are the default angle measurement.

SINH(z)

Returns *hyperbolic sin* of the argument.

Examples:

if $x = 1$ then $\sinh(x) = 1.75201$.

if $z = (3 + j * 2)$ then $\sinh(z) = [-4.168907; 9.154499]$.

SQR(z)

Returns the square root of the argument.

Examples:

if $x = -1$ then $sqr(x) = [0:1]$

if $z = [0:4]$ then $sqr(z) = [1.41421:1.41421]$

TAN(z)

Returns *tan* of the argument. The return value depends on the current default angle measurement [option](#). You can use the [TODEGREES](#) or the [TORADIANS](#) function to convert the argument to the desired angle measurement.

Examples:

if $x = 45$ then $\tan(x) = 1$ if degrees are the default angle measurement.

if $z = [1:1]$ then $\tan(z) = [0.272:1.084]$ if radians are the default angle measurement.

TANH(z)

Returns *hyperbolic tan* of the argument.

Examples:

if $x = 1$ then $\tanh(x) = 0.76159$

if $z = [0:1]$ then $\tanh(z) = [0:0.01745]$.

TODEGREES(z)

Converts to the argument, assumed to be in RADIANS, to *degrees*.

Examples:

if $x = 3.14159$ then $todegrees(x) = 179.99985$

if $z = [0:4]$ then $todegrees(z) = [0:229.18312]$

TORADIANS(z)

Converts to the argument, assumed to be in DEGREES, to *radians*.

Examples:

if x = 180 then *toradians*(x) = 3.14159

if z = [0:4] then *toradians*(z) = [0:0.06981]

